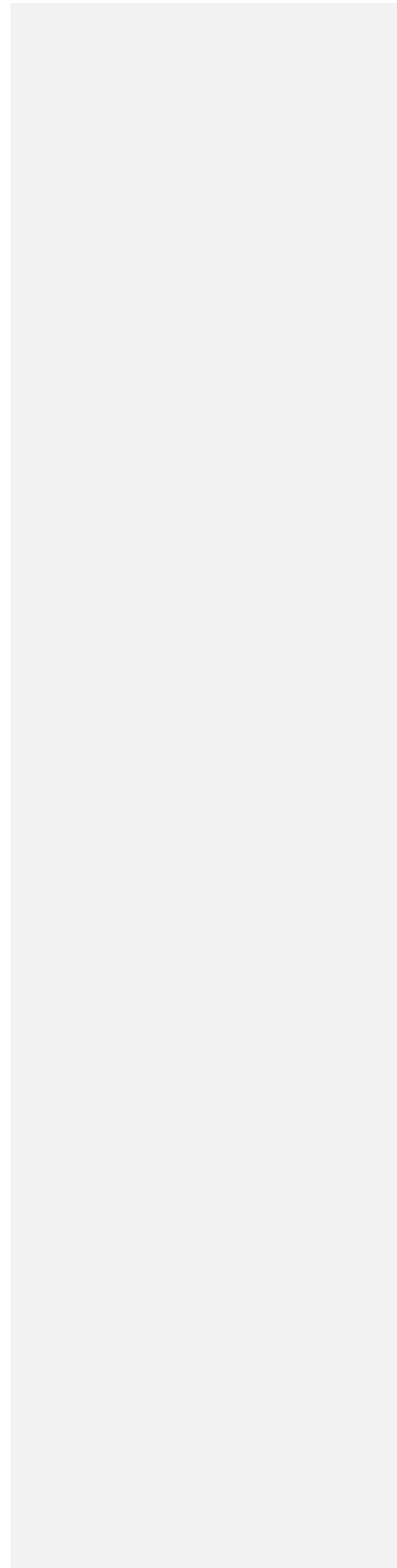


Processos, Qualidade e Gestão de Software

v3 23 nov 2009



Índice

1.1 INTRODUÇÃO	40
1.2.2 O RUP E SUAS CARACTERÍSTICAS	42
1.2.3 VISÃO GERAL DO RUP	42
1.2.4.1 CONCEPÇÃO	43
1.2.4.2 ELABORAÇÃO	44
1.2.4.3 CONSTRUÇÃO	45
1.2.4.4 TRANSIÇÃO	45
• O LIVRO INTRODUÇÃO AO RUP: RATIONAL UNIFIED PROCESS DO AUTOR PHILIPPE KRUCHTEN;	58
2.1 INTRODUÇÃO A PROCESSOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE	103
2.2 O MANIFESTO ÁGIL	104
2.3 PRINCIPAIS PROCESSOS ÁGEIS	105
2.4 EXTREME PROGRAMMING (XP)	106
2.4.1 VALORES DO XP	106
2.4.2 PRINCÍPIOS DO XP	107
2.4.3 PRÁTICAS DO XP	109
2.4.4 PAPÉIS DO XP	112
2.4.5 CICLO DE VIDA DO PROJETO XP	113
2.6. SCRUM	115
2.6.1 CARACTERÍSTICAS DO SCRUM	115
3.6.2 PAPÉIS DO SCRUM	115
2.6.3 PRÁTICAS DO SCRUM	116
2.6.4 CICLO DE VIDA DO SCRUM	117
2.6 FEATURE DRIVEN DEVELOPMENT	117
2.6.1 CARACTERÍSTICAS DO FDD	117
2.6.2 PAPÉIS DO FDD	118
2.6.3 PRÁTICAS DO FDD	119
2.6.7 CICLO DE VIDA DO FDD	120

2.7. TÓPICOS DE PESQUISA	122
2.8. SUGESTÕES DE LEITURA	122
2.9. EXERCÍCIOS	123
REFERÊNCIAS	123
3.1 INTRODUÇÃO	2
3.2 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE	3
3.2.1 MOTIVAÇÕES PARA O DDS	3
3.2.1 NÍVEIS DE DISPERSÃO	4
3.2.2 MODELOS DE NEGÓCIO	5
3.2.3 DESAFIOS	6
<ul style="list-style-type: none"> • <u>ARQUITETURA DO SOFTWARE: É UM DOS FATORES MAIS UTILIZADOS PARA A DIMINUIÇÃO DO ESFORÇO ENTRE AS EQUIPES. CONFORME KAROLAK [1998], UMA ARQUITETURA APROPRIADA PARA O DDS DEVE SE BASEAR NO PRINCÍPIO DA MODULARIDADE, POIS PERMITE ALOCAR TAREFAS COMPLEXAS DE FORMA DISTRIBUÍDA. COM ISSO HÁ UMA REDUÇÃO NA COMPLEXIDADE E É PERMITIDO UM DESENVOLVIMENTO EM PARALELO SIMPLIFICADO.</u> 	6
<ul style="list-style-type: none"> • <u>ENGENHARIA DE REQUISITOS: A ENGENHARIA DE REQUISITOS CONTÉM DIVERSAS TAREFAS QUE NECESSITAM DE ALTO NÍVEL DE COMUNICAÇÃO E COORDENAÇÃO. COM ISSO OS PROBLEMAS APRESENTADOS SÃO MAIS COMPLEXOS EM UM CONTEXTO DE DDS.</u> 	6
<ul style="list-style-type: none"> • <u>GERÊNCIA DE CONFIGURAÇÃO: O GERENCIAMENTO DE CONFIGURAÇÃO (CM) É A CHAVE PARA CONTROLAR AS MÚLTIPLAS PEÇAS EM UM PROJETO DISTRIBUÍDO. CONTROLAR MODIFICAÇÕES NOS ARTEFATOS EM CADA UMA DAS LOCALIDADES COM O PROCESSO DE DESENVOLVIMENTO DE TODO PRODUTO PODE SER COMPLEXO. APESAR DA UTILIZAÇÃO DE PRÁTICAS DE CM AUXILIAR NO CONTROLE DA DOCUMENTAÇÃO E DO SOFTWARE, A GERÊNCIA DE MODIFICAÇÕES SIMULTÂNEAS A PARTIR DE LOCAIS DIFERENTES É UM GRANDE DESAFIO. ALÉM DISSO, O USO DE FERRAMENTAS DE CM COMPARTILHADAS POR DUAS OU MAIS EQUIPES DE FORMA INADEQUADA GERA DIVERSOS RISCOS E PROBLEMAS EM PROJETOS DDS.</u> 	6
<ul style="list-style-type: none"> • <u>PROCESSO DE DESENVOLVIMENTO: EM PROJETOS DDS, O USO DE UMA METODOLOGIA QUE AUXILIA A SINCRONIZAÇÃO DAS ATIVIDADES É ESSENCIAL. COM ISSO TODOS OS MEMBROS UTILIZAM UMA NOMENCLATURA COMUM EM SUAS ATIVIDADES.</u> 	7
<ul style="list-style-type: none"> • <u>TESTES :</u> 	7
3.3 PROCESSOS PARA DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE	7
<u>EM UM AMBIENTE DE DESENVOLVIMENTO DISTRIBUÍDO, UM PROCESSO DE DESENVOLVIMENTO COMUM À EQUIPE É FUNDAMENTAL, TENDO EM VISTA QUE UMA METODOLOGIA AUXILIA DIRETAMENTE NA SINCRONIZAÇÃO, FORNECENDO AOS MEMBROS DA EQUIPE UMA NOMENCLATURA COMUM DE TAREFAS E</u>	

ATIVIDADES, E UM CONJUNTO COMUM DE EXPECTATIVAS AOS ELEMENTOS ENVOLVIDOS NO PROCESSO [PRIKLADNICKI 2008]. 7

A ENGENHARIA DE SOFTWARE (ES) SEMPRE ESTÁ APRESENTANDO GRANDES AVANÇOS E TRANSFORMAÇÕES RELACIONADAS ÀS TÉCNICAS, MODELOS E METODOLOGIAS. ESSES AVANÇOS SÃO DESTACADOS QUANDO SE TRABALHA COM PROCESSO DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE (DDS), HAVENDO UMA NECESSIDADE DO USO DE PRÁTICAS QUE DÊ SUPORTE ÀS DIFICULDADES ENCONTRADAS NAS DEFINIÇÕES DE REQUISITOS QUE MUDAM DE FORMA DINÂMICA NO DECORRER DO TEMPO. ESTUDOS RELACIONADOS A PROCESSO PARA DDS AINDA É ESCASSO, SENDO ASSIM ESTE CAPÍTULO RELATA O USO DE PRÁTICAS DO DESENVOLVIMENTO TRADICIONAL QUE PODEM SER IMPLANTADAS EM UM AMBIENTE DISTRIBUÍDO E AS POSSÍVEIS ADAPTAÇÕES. 7

3.4 PROCESSOS E ADAPTAÇÃO DAS PRÁTICAS EM PROJETOS DDS 7

A FORMA COMO UM PRODUTO DE SOFTWARE É CONCEBIDO, DESENVOLVIDO, TESTADO E ENTREGUE AO CLIENTE SOFRE GRANDE IMPACTO QUANDO O AMBIENTE DE DESENVOLVIMENTO É DISTRIBUÍDO [HERBSLEB 2001]. ASSIM, A ESTRUTURA NECESSÁRIA PARA O SUPORTE DESSE TIPO DE DESENVOLVIMENTO SE DIFERENCIA DA UTILIZADA EM AMBIENTES CENTRALIZADOS. DIFERENTES CARACTERÍSTICAS E TECNOLOGIAS SE FAZEM NECESSÁRIAS, CRESCENDO A IMPORTÂNCIA DE ALGUNS DETALHES ANTES NÃO PERCEBIDOS. 7

3.4.1 MODELO DE KAROLAK 8

KAROLAK [1998] ABORDA O DDS SEGUINDO O CICLO DE VIDA TRADICIONAL DE UM PROJETO DE DESENVOLVIMENTO DE SOFTWARE. O AUTOR PROPÕE UM MODELO PARA DESENVOLVER PROJETOS DDS ABRANGENDO AS ATIVIDADES QUE DEVER OCORRER AO LONGO DO CICLO DE VIDA. A FIGURA ABAIXO ILUSTRA O MODELO PROPOSTO (FIGURA 3.2): 8

8

FIGURA 3.2 – MODELO PARA PROJETOS DDS [KAROLAK 1998] 8

3.4.2 Uso de Práticas Ágeis 10

3.4.2.1 DXP – *DISTRIBUTED EXTREME PROGRAMMING* 11

3.4.2.2 ADOÇÃO DE *SCRUM* EM UM AMBIENTE DDS 12

3.5 CONSIDERAÇÕES FINAIS 15

O DESENVOLVIMENTO DE SOFTWARE SEMPRE SE APRESENTOU DE FORMA COMPLEXA. EXISTE UMA SÉRIE DE PROBLEMAS E DESAFIOS INERENTES AO PROCESSO. ASSIM COMO O PROCESSO DE DESENVOLVIMENTO DE SOFTWARE TEM SE TORNADO CADA VEZ MAIS COMPLEXO, A DISTRIBUIÇÃO DAS EQUIPES NO TEMPO E NO ESPAÇO TEM TORNADO OS PROJETOS DISTRIBUÍDOS CADA VEZ MAIS COMUNS. O DDS, AO ACRESCENTAR FATORES COMO DISPERSÃO GEOGRÁFICA, DISPERSÃO TEMPORAL E DIFERENÇAS CULTURAIS, ACENTUARAM ALGUNS DOS DESAFIOS EXISTENTES E ACRESCENTOU NOVOS DESAFIOS AO PROCESSO DE DESENVOLVIMENTO. O TRABALHO EM AMBIENTES DE DDS É MAIS COMPLEXO DO QUE EM AMBIENTES CENTRALIZADOS E NÃO EXISTEM MÉTODOS OU PRÁTICAS ESPECÍFICAS PARA O AMBIENTE DISTRIBUÍDO. 15

3.6 TÓPICOS DE PESQUISA 15

3.7 SUGESTÕES DE LEITURA 16

3.8 EXERCÍCIOS 16

1. DEFINA O QUE É DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE. 16

2. QUAIS AS VANTAGENS QUE UMA ORGANIZAÇÃO TEM AO UTILIZAR UM PROCESSO DDS? 16

3. QUAIS SÃO OS NÍVEIS DE DISPERSÃO EM UM AMBIENTE DDS? EXEMPLIFIQUE. 16

4. QUAIS OS MODELOS DE NEGÓCIO EM UM AMBIENTE DDS? EXEMPLIFIQUE. 16

5. QUAIS AS PRINCIPAIS DIFICULDADES AO REALIZAR UM PROJETO DDS? 16

REFERÊNCIAS 18

HERBSLEB, J. D., MOITRA, D. "GLOBAL SOFTWARE DEVELOPMENT", IEEE SOFTWARE, MARCH/APRIL, EUA, 2001, P. 16-20. 18

KAROLAK, D. W. "GLOBAL SOFTWARE DEVELOPMENT – MANAGING VIRTUAL TEAMS AND ENVIRONMENTS". LOS ALAMITOS, IEEE COMPUTER SOCIETY, EUA, 1998, 159P. 18

KIEL, L. "EXPERIENCES IN DISTRIBUTED DEVELOPMENT: A CASE STUDY", IN: WORKSHOP ON GLOBAL SOFTWARE DEVELOPMENT AT ICSE, OREGON, EUA, 2003, 4P. 18

KIRCHER, M., JAIN, P., LEVINE, A. "DISTRIBUTED EXTREME PROGRAMMING", IEEE, AGILE 2008. 18

HERBSLEB, J.D., MOCKUS, A., FINHOLT, T.A. E GRINTER, R. E. "AN EMPIRICAL STUDY OF GLOBAL SOFTWARE DEVELOPMENT: DISTANCE AND SPEED", IN: ICSE 2001, TORONTO, CANADA. 18

CARMEL, E. "GLOBAL SOFTWARE TEAMS – COLLABORATING ACROSS BORDERS AND TIME-ZONES" PRENTICE HALL, EUA, 1999, 269P. 18

MARQUARDT, M. J., HORVATH, L. "GLOBAL TEAMS: HOW TOP MULTINATIONALS SPAN BOUNDARIES AND CULTURES WITH HIGH-SPEED TEAMWORK". DAVIES-BLACK. PALO ALTO, EUA, 2001. 18

<u>YOUNG, C., TERASHIMA, H. "HOW DID WR ADAPT AGILE PROCESSES TO OUR DISTRIBUTED DEVELOPMENT?"</u> <u>IEEE, AGILE 2008.</u>	<u>18</u>
<u>PRIKLADNICKI, R., AUDY, J. L. N., EVARISTO, R. "GLOBAL SOFTWARE DEVELOPMENT IN PRACTICE: LESSONS LEARNED", JOURNAL OF SOFTWARE PROCESS: PRACTICE AND IMPROVEMENT – SPECIAL ISSUE ON GLOBAL SOFTWARE DEVELOPMENT, 2004.</u>	<u>18</u>
<u>PRIKLADNICKI, R. "MUNDDOS: UM MODELO DE REFERÊNCIA PARA DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE". DISSERTAÇÃO DE MESTRADO, PPGCC – PUCRS, BRASIL, 2003.</u>	<u>18</u>
<u>J. L. N. PRIKLADINICKI, R.; AUDY. DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE. 2007.</u>	<u>18</u>
<u>PERRELLI, HERMANO. VISÃO GERAL DO RUP. CENTRO DE INFORMÁTICA, UNIVERSIDADE FEDERAL DE PERNAMBUCO. DISPONÍVEL EM: HTTP://WWW.CIN.UFPE.BR/~IF717/SLIDES/3-VISAO-GERAL-DO-RUP.PDF. ACESSADO EM 20 MAIO 2009.</u>	<u>18</u>
<u>PRESSMAN, ROGER S. SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH. EUA: MCGRAW HILL, 2001. 860 P.</u>	<u>18</u>
<u>SOMMERVILLE, IAN. SOFTWARE ENINNERING. 8.ED. [S.L] ADDISON WESLEY, 2007.</u>	<u>18</u>
<u>SUTHERLAND, J., "DISTRIBUTED SCRUM: AGILE PROJECT MANAGEMENT WITH OUTSOURCED DEVELOPMENT TEAMS", HICSS, 2007.</u>	<u>19</u>
<u>TELES, VINÍCIUS MANHÃES. EXTREME PROGRAMMING: APRENDA COMO ENCANTAR SEUS USUÁRIOS DESENVOLVENDO SOFTWARE COM AGILIDADE E ALTA QUALIDADE. 1. ED. SÃO PAULO: NOVATEC, 2004. 320 P.</u>	<u>19</u>
<u>TRAVASSOS, G. H., ABRANTES, J. F., "CARACTERIZAÇÃO DE MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE"</u>	<u>19</u>
<u>10.1 INTRODUÇÃO</u>	<u>22</u>
<u>10.2 ARQUITETURA DIRIGIDA A MODELOS</u>	<u>22</u>
<i>10.2.1. CONCEITOS BÁSICOS</i>	<i>23</i>
<i>10.2.2. PADRÕES OMG E A ARQUITETURA MDA</i>	<i>29</i>
<u>10.3 ABORDAGENS MDD MODELOS</u>	<u>31</u>
<i>10.3.1 OO-METHOD</i>	<i>31</i>
10.3.1.1. O PROCESSO BÁSICO DE TRANSFORMAÇÃO	31
10.3.1.2. COMPARAÇÃO COM MDA	32
10.3.1.3. O MODELO CONCEITUAL	33
10.3.1.4. O COMPILADOR DE MODELOS	35
10.3.1.5. OLIVANOVA	35
<i>10.3.2 . ANDROMDA</i>	<i>35</i>

10.4 PROBLEMAS E DESAFIOS DOS PROCESSOS MDD	36
<i>10.4.1. VISÃO GERAL</i>	36
<i>10.4.2. LIÇÕES APRENDIDAS NA ADOÇÃO DE SOLUÇÕES MDA</i>	37
<i>10.4.3. O PROGRAMA FASTSTART DA OMG</i>	37
10.5. TÓPICOS DE PESQUISA	38
10.6 . SUGESTÕES DE LEITURA	38
10.7 . EXERCÍCIOS	39
REFERÊNCIAS	41
1. INTRODUÇÃO	42
1.1. O QUE É MODELAGEM DE PROCESSOS	42
2.1. OBJETIVO DA MODELAGEM DE PROCESSOS SOFTWARE	45
3.1 VANTAGENS E DESVANTAGENS DA UTILIZAÇÃO DE MODELAGEM DE PROCESSOS	45
A. VANTAGENS	45
• BONS MODELOS DE PROCESSOS, SÃO A CHAVE PARA A BOA COMUNICAÇÃO.	46
• SE O PROCESSO, É ALGUMA COISA NOVA QUE A EMPRESA ESTÁ PLANEJANDO EXECUTAR, O MODELO PODE AJUDAR A ASSEGURAR SUA EFICIÊNCIA DESDE O INÍCIO.	46
• REVELAR ANOMALIAS, INCONSISTÊNCIAS, INEFICIÊNCIAS E OPORTUNIDADES DE MELHORIA, PERMITINDO À ORGANIZAÇÃO QUE SE COMPREENDA MELHOR E AUXILIANDO NA REENGENHARIA DESSES PROCESSOS.	46
• FORNECER VISÃO CLARA E UNIFORMIZADA DAS ATIVIDADES, SUAS RAZÕES E FORMAS DE EXECUÇÃO.	46
• UTILIZAR O MODELO COMO UM MEIO PARA DISTRIBUIÇÃO DE CONHECIMENTO DENTRO DA ORGANIZAÇÃO E TREINAR AS PESSOAS, AJUDANDO-AS A CONHECER MELHOR SEUS PAPÉIS E AS TAREFAS QUE EXECUTAM.	46
B. DESVANTAGENS	46
4.1. LINGUAGEM DE MODELOS DE PROCESSOS	46
4.1.1. BPM	46
4.1.2. SPEM	53
4.2.1 OMG(OBJECT MANAGEMENT GROUP)	58
COMMON WAREHOUSE METAMODEL™	59
COMMON OBJECT REQUEST BROKER ARCHITECTURE®	59
6.1. FERRAMENTAS DE MODELAGEM	59



6.1.1. COMPARAÇÃO ENTRE AS FERRAMENTAS	62
7.1. SUGESTÕES DE LEITURA	63
8.1. TÓPICOS DE PESQUISA	63
EXERCÍCIO	64
REFERENCIAS	64
6.1. INTRODUÇÃO	68
6.2. O QUE É QUALIDADE?	68
6.3. COMPETITIVIDADE X PRODUTIVIDADE	69
6.3.1. CONCEITO DE PRODUTIVIDADE	69
6.3.2. CONCEITO DE COMPETITIVIDADE	70
6.4. QUALIDADE TOTAL	71
6.4.1. DEMING	71
6.4.2. JURAN	72
6.4.3. CROSBY	72
6.4.4. FEIGENBAUN	73
6.4.5. ISHIKAWA	73
6.5. CONTROLE DA QUALIDADE TOTAL	74
6.5.1. APRESENTAÇÃO DO CONTROLE DA QUALIDADE TOTAL	76
6.5.2. SIGNIFICADO DO CONTROLE DA QUALIDADE TOTAL	76
6.5.3. PRINCÍPIOS DA QUALIDADE TOTAL	77
6.6. CONTROLE DE PROCESSO	80
6.6.1 CONCEITO DE PROCESSO	80
6.6.2 CONCEITO DE CONTROLE DE PROCESSO	80
6.6.3 MÉTODO DE CONTROLE DE PROCESSO	81

6.7. FERRAMENTAS DA QUALIDADE	82
6.7.1. AS SETE FERRAMENTAS DA QUALIDADE	82
6.8. GESTÃO DA QUALIDADE	86
6.8.1. GERENCIAMENTO POR DIRETRIZES	86
6.8.2. GERENCIAMENTO DA ROTINA	86
6.9. GARANTIA DA QUALIDADE	86
6.10. QUALIDADE NA INTERFACE COMPRAS/VENDAS	87
6.10.1. QUALIDADE NAS VENDAS	87
6.10.2. QUALIDADE NAS COMPRAS	89
6.11. IMPLANTAÇÃO DO TQC	91
6.11.1 FUNDAMENTOS	91
6.11.2 ORGANIZAÇÃO PARA IMPLANTAÇÃO	91
6.11.3 SISTEMA DE GERENCIAMENTO DA IMPLANTAÇÃO DO TQC	92
6.12 TÓPICOS DE PESQUISA	93
6.13 SUGESTÕES DE LEITURA	93
6.14 EXERCÍCIOS	93
6.15 REFERÊNCIAS	94
7.2. ORGANISMOS NORMATIVOS	102
7.2.1 ISO	103
A PRIMEIRA VERSÃO ISO 9000:1987 SUBDIVIDIA-SE EM MODELOS PARA QUALIDADE, CLASSIFICADOS DA SEGUINTE FORMA [MATOS 2009]:	106
<ul style="list-style-type: none"> ● ISO 9001 : MODELO DE GARANTIA PARA QUALIDADE DE PROJETO, DESENVOLVIMENTO, PRODUÇÃO, MONTAGEM E FORNECEDORES APLICANDO-SE À ORGANIZAÇÕES CUJAS ATIVIDADES ERAM VOLTADAS PARA CRIAÇÃO DE NOVOS PRODUTOS. 	106
7.3.4. NORMA ISO 9001	109
COM O LANÇAMENTO DA ISO 9000, VÁRIAS ORGANIZAÇÕES DESPERTARAM A TEMÁTICA DE QUE PRECISAVAM IMPOR, E PRINCIPALMENTE MANTER, PADRÕES DE QUALIDADE EM SEU FUNCIONAMENTO, SEJA NOS PROCESSOS, OU MESMO NAS PESSOAS QUE COLABORAM PARA O FUNCIONAMENTO DAS MESMAS. O	

PENSAMENTO COM UMA MELHOR VISÃO E AMBIÇÃO PARA O MERCADO DISPÕE DA REALIZAÇÃO DE INVESTIMENTOS QUE PRESTEM ALTERNATIVAS VIÁVEIS PARA O CRESCIMENTO E MELHORAMENTO DAS ATIVIDADES. 109

MELLO ET AL. (2009) DESCREVE QUE AS NORMAS PARA SISTEMAS DE GESTÃO, PRINCIPALMENTE A ISO 9001, FORNECEM MODELOS BÁSICOS PARA QUE AS ORGANIZAÇÕES PREPAREM E OPEREM SEUS FLUXOS DE FUNCIONAMENTO SEGURAMENTE FORTIFICADOS. O AUTOR AINDA CITA QUE: 109

“AS GRANDES ORGANIZAÇÕES, OU AQUELAS COM PROCESSOS COMPLEXOS, PODERIAM NÃO FUNCIONAR BEM SEM UM SISTEMA DE GESTÃO, APESAR DE ELE PODER TER SIDO CHAMADO POR ALGUM OUTRO NOME.” 109

A NORMA ISO 9001 FOI INSTITUÍDA COM ESSE PROPÓSITO. DESCREVER OS REQUISITOS PARA POSSIBILITAR A IMPLANTAÇÃO E ADMINISTRAÇÃO DE UM MODELO PARA GARANTIA DE QUALIDADE PARA PRODUTOS E SERVIÇOS ATRAVÉS DE UM SISTEMA DE GESTÃO DE QUALIDADE. COMO ESTRATÉGIA DE NEGÓCIOS PARA APRESENTAR UMA BASE SÓLIDA DE SEGURANÇA E QUALIDADE NAS EMPRESAS, ESTA NORMA CONDIZ UM FATOR DE CERTIFICAÇÃO ATRAVÉS DE AUDITORIAS, INSPEÇÕES, DENTRE OUTRAS ATIVIDADES QUE CLASSIFIQUEM E GARANTAM BOA PROCEDÊNCIA PARA VERIFICAÇÃO E VALIDAÇÃO DE PROCESSOS E SERVIÇOS CONFORME AS TERMINOLOGIAS E VOCABULÁRIOS APRESENTADOS PELA ISO NA VERSÃO 9000. 109

7.5 ISO/IEC 15504 123

7.5.5 DIMENSÃO DE CAPACIDADE 128

7.6 CONCLUSÕES 130

O ESTUDO DA MELHORIA DE PROCESSOS COM O USO DE NORMAS ISO É COMUMENTE RELEVANTE A COMPLEXIDADE EXISTENTE NO CONTEXTO DE ABORDAGEM CARACTERIZADOR DAS DITAS NORMAS. OS PADRÕES FIXOS DELIMITAM A ADEQUAÇÃO DO USO DESTES ARTIFÍCIOS, QUE COMPREENDEM RETORNOS GARANTIDOS QUANDO APLICADOS CORRETAMENTE, APESAR DOS CUSTOS E INVESTIMENTOS QUE SE FAZEM NECESSÁRIOS NO ÂMBITO DA OBTENÇÃO DE UM RECONHECIMENTO QUALITATIVO INDISPENSÁVEL PARA OS PROJETOS DE SISTEMAS NAS ORGANIZAÇÕES. 130

MESMO COM TANTOS EMPECILHOS, PRINCIPALMENTE PARA AS ORGANIZAÇÕES DE PEQUENO E MÉDIO PORTE, O USO DAS NORMAS CRESCE A CADA DIA E OS RESULTADOS OBTIDOS DESPERTAM A ATENÇÃO DO MERCADO PARA A VALORIZAÇÃO DOS PRINCÍPIOS DE RESPONSABILIDADE, ÉTICA, COMPROMISSO, E PRINCIPALMENTE RESPEITO AO CLIENTE, TRAZENDO VALORES QUE ANTES NÃO ERAM RECONHECIDOS DEVIDO A FALTA DE UMA ADMINISTRAÇÃO CONSCIENTE QUE DESMITIFICASSE O CONCEITO DE AUTO-SUFICIÊNCIA E TRABALHASSE REALMENTE NA EXPECTATIVA DE VALORIZAR QUEM TRAZ CONSIGO UM PONTO DEFINIDOR PARA O PROGRESSO OU FRACASSO DOS PRODUTOS E SERVIÇOS POR ELAS GERADAS. 130

ESTE CAPÍTULO BUSCOU DE FORMA SUCINTA E OBJETIVA APRESENTAR OS PRINCIPAIS CONCEITOS DE ALGUMAS DAS PRINCIPAIS NORMAS PARA QUALIDADE DE PROCESSOS DE SOFTWARE UTILIZADAS ATUALMENTE. FORAM ABORDADAS AS ENTIDADES QUE REGEM ESSAS NORMAS, A ISO, O IEC E A ABNT, BREVES HISTÓRIAS DE CADA UMA DELAS, AS RELAÇÕES QUE AS NORMAS POSSUEM COM OUTROS MODELOS DE MELHORIAS DE PROCESSOS, UMA ABORDAGEM SIMPLES E DIRETA DE SISTEMAS DE GESTÃO PARA QUALIDADE, ALÉM DO CICLO DE VIDA PARA O DESENVOLVIMENTO DE SOFTWARE E ALGUNS MODELOS DE ACOMPANHAMENTO E AVALIAÇÃO DE PROCESSOS PARA OBTENÇÃO DE QUALIDADE EM PROCESSOS DE SOFTWARE. 130

SUGESTÕES DE LEITURA **130**

PARA SE OBTER MAIS INFORMAÇÕES SOBRE AS NORMAS ABORDADAS NESTE CAPÍTULO ESTÃO DESCRITOS ALGUNS LIVROS E DOCUMENTOS ESSENCIAIS DE AUTORES RENOMADOS SOBRE OS RESPECTIVOS ASSUNTOS: **130**

• **NORMA ISO 9001: 2008: SISTEMA DE GESTÃO DA QUALIDADE PARA OPERAÇÕES DE PRODUÇÃO E SERVIÇOS DE MELLO & SILVA 2009** **130**

• **NORMAS ISO 9001, ISO/IEC 12207 E ISO/IEC 15504: QUALIDADE DE SOFTWARE: APRENDA AS METODOLOGIAS E TÉCNICAS MAIS MODERNAS PARA O DESENVOLVIMENTO DE SOFTWARE. 2 ED. DE KOSCIANSKI E SOARES 2007** **130**

• **SISTEMAS DE GESTÃO PARA QUALIDADE DE PROCESSOS: GESTÃO DA QUALIDADE 9ª EDIÇÃO MARSHALL JÚNIOR ET AL. 2009** **130**

TÓPICOS DE PESQUISA **132**

AS PESQUISAS SOBRE OS TÓPICOS ABORDADOS NESTE CAPÍTULO SÃO DIVERSIFICADAS PELAS INSTITUIÇÕES MANTENEDORAS DOS PADRÕES DAS NORMAS. VISTO QUE SÃO CONTEÚDOS RESTRITOS A CONTRATOS E FECHADOS PARA CONSULTA PÚBLICA, OS AVANÇOS NAS PESQUISAS SE DÃO APENAS DE ATUALIZAÇÕES DAS CORRENTES NORMAS PELA ISO PROVENIENTES DE EVENTOS, ENCONTROS, CONGRESSOS, DENTRE OUTRAS OPORTUNIDADES, ONDE SÃO REPASSADAS EXPERIÊNCIAS DE VÁRIOS ÓRGÃOS, SEJAM PRIVADOS OU NÃO, ESTIPULANDO QUAIS AS PRINCIPAIS DIFICULDADES, DIRETRIZES, RESTRIÇÕES QUE PRECISAM SER ATUALIZADAS. **132**

NO BRASIL A PESQUISA SEGUE OS MESMOS RIGOROSOS CRITÉRIOS INTUÍDOS PELA ABNT. VÁRIOS ASSOCIADOS FORMAM O COMITÊ DA QUALIDADE NÚMERO VINTE E CINCO ([HTTP://WWW.ABNTCB25.COM.BR/](http://www.abntcb25.com.br/)) REGENDO AS NORMAS NO FORMATO NBR E AUXILIANDO AS EMPRESAS DE CONSULTORIAS NO PROCESSO DE CERTIFICAÇÃO E IMPLANTAÇÃO DE NORMAS ISO PARA GESTÃO DA QUALIDADE. **132**

EXERCÍCIOS **132**

1. AO LONGO DE TODO O CAPÍTULO TORNA-SE NOTÁVEL A IMPORTÂNCIA QUE AS NORMAS EXERCEM NO CONTEXTO DOS PADRÕES QUE DEVEM SER ADOTADOS PELAS AS EMPRESAS PARA QUE AS MESMAS SE DESTAQUEM NO MERCADO QUE DEMANDA MAIOR QUALIDADE E PRATICIDADE E MENOR TEMPO E CUSTO. QUAL A IMPORTÂNCIA DE PROPOR A ADOÇÃO DE NORMAS ISO SEJA EM CARÁTER CERTIFICADOR OU NÃO, NOS DIAS ATUAIS? **132**

2. A IMPLANTAÇÃO E MANUTENÇÃO DE SISTEMAS DE GESTÃO PARA QUALIDADE EM ORGANIZAÇÕES COM NORMAS ISO, DENTRE ELAS A ISO 9001, ENVOLVE UM MACRO PLANEJAMENTO DESDE A ALTA HIERARQUIA AOS COLABORADORES TÉCNICOS CONFORME APRESENTADOS NA SEÇÃO 7.3.1 EM QUESTÃO. SABE-SE QUE A DEFINIÇÃO DE UM SISTEMA ÚNICO E PADRONIZADO ENVOLVE TODOS OS PROCESSOS, SUAS ATIVIDADES E TAREFAS, ALÉM DA COMPLETA DEDICAÇÃO DOS PROFISSIONAIS PARA COM SEU CORRETO FUNCIONAMENTO. ESBOCE UM PEQUENO ÍNDICE DE PRÁTICAS QUE PODERIAM SER AGREGADAS AOS OITO PRINCÍPIOS DA VERSÃO ISO 9000:2000 MENCIONADAS NA MESMA SEÇÃO QUE TRARIAM MELHORIAS SIGNIFICATIVAS DURANTE A IMPLANTAÇÃO DE UM SGQ. **132**

3. BASEADO NAS INFORMAÇÕES APRESENTADAS NESTE CAPÍTULO, NA SEÇÃO 7.3.6, DESCREVA RESUMIDAMENTE COM SUAS PALAVRAS O PROCESSO DE CONSULTORIA E IMPLANTAÇÃO DA ISO 9001:2008.	132
4. A ISO/IEC 90003 É UM GUIA TÉCNICO COMPLEMENTAR A ISO 9001 PARA FÁBRICAS DE SOFTWARE. EXPLIQUE CADA ATIVIDADE, DE CICLO DE VIDA E SUPORTE, CITANDO SUAS CARACTERÍSTICAS E PRINCIPAIS DIRETRIZES PARA IMPLANTAÇÃO.	132
5. O QUE VOCÊ ENTENDE POR “CICLO DE VIDA DE DESENVOLVIMENTO” SEGUNDO A NORMA ISO/IEC 12207?	132
6. O PROCESSO DE ADAPTAÇÃO DA NORMA ISO/IEC 12207 ENVOLVE ALGUMAS PRÁTICAS ADMINISTRATIVAS ESSENCIAIS QUE TODAS AS ORGANIZAÇÕES DEVERIAM ADOTAR NO SEU FLUXO DE FUNCIONAMENTO INDEPENDENTE DA IMPLANTAÇÃO DA NORMA. DESENVOLVA UM ESBOÇO QUE ASSOCIE ESSAS PRÁTICAS AOS PROCESSOS PRIMÁRIOS, ORGANIZACIONAIS E DE APOIO.	132
7. O PROJETO SPICE SURTIU COM O INTUITO DE AMENIZAR AS DIFICULDADES DE IMPLANTAÇÃO PROVENIENTES DO CMMI E OUTROS MODELOS MAIS COMPLEXOS DA ÉPOCA QUE ERAM VOLTADOS PARA GRANDES ORGANIZAÇÕES E EXIGIAM ALTOS CONHECIMENTOS SOBRE OS CONCEITOS DE PROCESSOS E SISTEMAS DE GESTÃO.FAÇA UMA PESQUISA QUE DESCREVA OS PROPÓSITOS PRINCIPAIS ALMEJADOS PELOS ENGENHEIROS DA ÉPOCA FAZENDO UM COMPARATIVO DESTE PROJETO COM AS PRINCIPAIS CARACTERÍSTICAS QUE DESCREVEM OS NÍVEIS DE MATURAÇÃO DO CMMI.	133
8. EXPLIQUE E DIFERENCIE OS MODELOS DE REFERÊNCIA PRM E PAM.	133
9. O QUE VOCÊ ENTENDE POR DIMENSÃO DE PROCESSO? ESTA DIMENSÃO TORNA-SE SEMELHANTE AOS CONCEITOS DE CICLO DE VIDA DA ISO/IEC 12207? APONTE UMA RELAÇÃO QUE PODERIA SER APRESENTADA BASEADA NESTAS DUAS NORMAS.	133
10. A DIMENSÃO DE CAPACIDADE PARA PROCESSOS, E OS NÍVEIS DE CAPACIDADE RELACIONADOS COM ESTA DIMENSÃO, EXERCEM TOTAL INFLUENCIA NAS ALTERAÇÕES DE CARACTERÍSTICAS DOS ATRIBUTOS E ATIVIDADES EXECUTADAS PELOS COLABORADORES. QUAIS FATORES INFLUENCIAM DIRETAMENTE NA PASSAGEM DE NÍVEL GRADATIVA QUE OS PROCESSOS ALCANÇAM A MEDIDA QUE SÃO AVALIADOS E MELHORADOS?	133
INTRODUÇÃO	102
HISTÓRICO	103
CMMI	103
REPRESENTAÇÕES DO MODELO CMMI	105
REPRESENTAÇÃO POR ESTÁGIOS	105
REPRESENTAÇÃO CONTÍNUA	107
REPRESENTAÇÃO POR ESTÁGIOS X CONTÍNUA	109
MÉTODO DE AVALIAÇÃO DO CMMI (SCAMPI)	109
8.3.2.1. CONCEITO CENTRAL	110
8.3.2.2. PARÂMETROS OBSERVADOS NO SCAMPI	110

8.3.2.3.	PRAZO E EXIGÊNCIA DE PESSOAL	110
8.3.2.4.	CARACTERÍSTICAS ESSENCIAIS DO MÉTODO DE SCAMPI	110
8.3.2.5.	MODOS DE USO	111
8.3.2.6.	DESCRIÇÃO DO MÉTODO	111
MPS.BR		115
8.4.1.	REPRESENTAÇÃO DO MODELO MPS	116
8.4.1.1.	NÍVEL G – PARCIALMENTE GERENCIADO	116
8.4.1.2.	NÍVEL F – GERENCIADO	117
8.4.1.3.	NÍVEL E – PARCIALMENTE DEFINIDO	117
8.4.1.4.	NÍVEL D – LARGAMENTE DEFINIDO	118
8.4.1.5.	NÍVEL C – DEFINIDO	118
8.4.1.6.	NÍVEL B – GERENCIADO QUANTITATIVAMENTE	118
8.4.1.7.	NÍVEL A – EM OTIMIZAÇÃO	118
8.4.2.	MÉTODO DE AVALIAÇÃO DO MPS.BR (MA-MPS)	118
8.4.2.1.	PRAZO E EXIGÊNCIA DE PESSOAL	120
8.4.2.2.	DESCRIÇÃO DO MÉTODO	120
CMMI X MPS.BR		123
EXERCÍCIOS		124
SUGESTÕES DE LEITURA		125
TÓPICOS DE PESQUISA		125
REFERÊNCIAS		126
INTRODUÇÃO A MODELOS PARA MELHORIA DE PROCESSOS DE SOFTWARE IDEAL		129
IDEAL		130
▪	FASES DO IDEAL	132
•	FASE INICIAL (INITIATING)	133
•	FASE DE DIAGNÓSTICO (DIAGNOSING)	135
•	FASE DE ESTABILIZAÇÃO (DIAGNOSING)	137
•	FASE DE AÇÃO (ACTING)	139
•	FASE DE APROVEITAMENTO (LEVERAGING)	141
•	FASE DE GERENCIAMENTO DO PROGRAMA DE MELHORIA DO PROCESSO DE SOFTWARE (MANAGE)	143
PRO2PI		145
▪	ENGENHARIA DE PROCESSO DIRIGIDA POR PERFIS DE CAPACIDADE E SEUS FUNDAMENTOS	146
▪	O PRO2PI	147
•	PRO2PI-PROP: PROPRIEDADES DE PRO2PI	152
•	PRO2PI-MODEL: MODELO DE PRO2PI	153
•	PRO2PI-MEAS: MEDIÇÕES PARA PRO2PI	155
•	PRO2PI-CYCLE: PROCESSO PARA CICLO DE MELHORIA	156
SEIS SIGMA		157
▪	PDCA	160
▪	DMAIC	160

• DEFINIR	161
• MEDIÇÃO	161
• ANÁLISE	162
• MELHORIA	162
• CONTROLE	162
CONSIDERAÇÕES FINAIS	163
EXERCÍCIOS	164
SUGESTÕES DE LEITURA	164

<u>PARA ENTENDER MELHOR O QUE É MELHORIA DE PROCESSO DE SOFTWARE LEIA A NORMA ISO/IEC 15504-4/2004.</u>	164
---	-----

<u>PARA UM ESTUDO DETALHADO SOBRE O PRO2PI LEIA TESE DE DOUTORADO DE CLÊNIO SALVIANO, “UMA PROPOSTA ORIENTADA A PERFIS DE CAPACIDADE DE PROCESSO PARA EVOLUÇÃO DA MELHORIA DE PROCESSO DE SOFTWARE”. DISPONÍVEL EM: HTTP://LIBDIGI.UNICAMP.BR/DOCUMENT/?CODE=VTL000380495</u>	164
--	-----

<u>PARA UM ESTUDO DETALHADO SOBRE IDEAL LEIA O GUIA OFICIAL DE IMPLANTAÇÃO PRODUZIDO PELO SEI, “IDEAL - A USER'S GUIDE FOR SOFTWARE PROCESS IMPROVEMENT”. DISPONÍVEL EM: HTTP://WWW.SEI.CMU.EDU/LIBRARY/ABSTRACTS/REPORTS/96HB001.CFM</u>	164
--	-----

TÓPICOS DE PESQUISA	165
REFERÊNCIAS	166

<u>10.1. MODELOS DE QUALIDADE DE PRODUTO</u>	172
--	-----

<u>OS MODELOS DE QUALIDADE OBJETIVAM AVALIAR O PRODUTO DE SOFTWARE, SEGUNDO DIFERENTES ASPECTOS BASEADOS NA VISÃO DO USUÁRIO. PARA PADRONIZAR INTERNACIONALMENTE AS CARACTERÍSTICAS DE IMPLEMENTAÇÃO DO SOFTWARE, FORAM CRIADAS ALGUMAS NORMAS QUE SERÃO VISTAS A SEGUIR.</u>	172
---	-----

<u>10.1.1. ISO 9126</u>	172
-------------------------	-----

<u>10.1.1.1. DIRETRIZES PARA USO DA NORMA NBR ISO/IEC 9126-1</u>	172
--	-----

<u>10.1.1.2. CARACTERÍSTICAS E SUBCARACTERÍSTICAS DE QUALIDADE DE SOFTWARE</u>	172
--	-----

<u>10.1.2. ISO 12119</u>	174
--------------------------	-----

<u>10.1.3. ISO 14598</u>	176
--------------------------	-----

<u>É UM GUIA PARA AVALIAÇÃO DE PRODUTOS DE SOFTWARE, BASEADO NA UTILIZAÇÃO PRÁTICA DA NORMA ISO 9126, JÁ QUE ESTA DEFINE AS MÉTRICAS, CARACTERÍSTICAS E SUBCARACTERÍSTICAS DE QUALIDADE DE SOFTWARE [KOSCIANSKI & SOARES, 2007].</u>	176
--	-----

<u>10.1.4. PROJETO SQUARE</u>	178
-------------------------------	-----

<u>10.1.5. NORMA SQUARE</u>	178
-----------------------------	-----

NA REORGANIZAÇÃO DAS ANTIGAS NORMAS 9126 E 14598, O PROJETO SQUARE ADOTOU UMA DIVISÃO DE ASSUNTOS EM CINCO TÓPICOS ILUSTRADOS NA FIGURA 10.2. 178

10.2. TESTE DE SOFTWARE 180

10.2.1. ABORDAGENS DE TESTES 181

EXISTEM DUAS ABORDAGENS PRINCIPAIS DE TESTES: ABORDAGEM FUNCIONAL (“BLACK BOX” OU “CAIXA PRETA”) E ABORDAGEM ESTRUTURAL (“WHITE BOX” OU “CAIXA BRANCA”) [SOMMERVILLE 2004], [PRESSMAN 2002]. 181

• CAIXA PRETA: COMO O PRÓPRIO NOME JÁ SUGERE, NESTA ABORDAGEM O TESTADOR VISUALIZA O SOFTWARE COMO UMA CAIXA PRETA, OU SEJA, NÃO CONSIDERA A ESTRUTURA INTERNA DO PROGRAMA, DE QUE FORMA O CÓDIGO FOI IMPLEMENTADO OU QUE TECNOLOGIA FOI UTILIZADA, POR EXEMPLO. CONSIDERANDO OS DADOS DE ENTRADA, O OBJETIVO PRINCIPAL É OBSERVAR AS SAÍDAS GERADAS PELO SISTEMA E VERIFICAR SE ESTAS ESTÃO DE ACORDO COM O ESPERADO. A FIGURA 10.3 ILUSTRA ESTE TIPO DE ABORDAGEM. 181

• CAIXA BRANCA: DIFERENTEMENTE DA ABORDAGEM ANTERIOR, NESTE TIPO DE ABORDAGEM O TESTADOR ESTÁ INTERESSADO NO QUE ESTÁ ACONTECENDO “DENTRO DA CAIXA”. É CARACTERIZADA POR AVALIAR AS FUNCIONALIDADES INTERNAS DOS COMPONENTES DO SOFTWARE, BASEANDO-SE NO CÓDIGO FONTE E PROCURANDO EXERCITAR ESTRUTURAS DE CONTROLE E DE DADOS DO PROGRAMA. SENDO ASSIM, FAZ-SE NECESSÁRIO QUE O ANALISTA DE TESTES TENHA BOA HABILIDADE EM PROGRAMAÇÃO DE MODO A ENTENDER TODOS OS CAMINHOS LÓGICOS POSSÍVEIS. A FIGURA 10.4 ILUSTRA A ABORDAGEM ESTRUTURAL. 181

10.2.2. ESTÁGIOS DE TESTES 181

OS TESTES DE SOFTWARE NORMALMENTE SÃO EXECUTADOS EM DIFERENTES ESTÁGIOS DURANTE O CICLO DE VIDA DO DESENVOLVIMENTO DO SOFTWARE. DEPENDENDO DO OBJETIVO PRINCIPAL DO TESTE, QUATRO ESTÁGIOS SÃO CONHECIDOS [GRAHAM ET. AL 2007], [MYERS 2004]: 181

• TESTE DE UNIDADE: REALIZA TESTES EM COMPONENTES INDIVIDUAIS (MÓDULOS, PROGRAMAS, OBJETOS, CLASSES, ETC) DE FORMA A DETERMINAR SE CADA UM DELES, SEPARADAMENTE, ESTÁ SENDO EXECUTADO DE MANEIRA CORRETA. NORMALMENTE ESTES TESTES SÃO DE CAIXA BRANCA, REALIZADOS PELOS PRÓPRIOS DESENVOLVEDORES DO COMPONENTE. GERALMENTE UTILIZAM FERRAMENTAS QUE PROVÊEM UM SUPORTE ADICIONAL PARA CHECAR A CORRETUDE DO PROGRAMA, COMO FERRAMENTA DE *DEBUGGING* OU *FRAMEWORK* PARA TESTE UNITÁRIO. POR EXEMPLO, OS DEFEITOS ENCONTRADOS NESTE ESTÁGIO SÃO NORMALMENTE CORRIGIDOS DE IMEDIATO, SEM A NECESSIDADE DE DOCUMENTÁ-LOS FORMALMENTE, E ASSIM, REDUZINDO O CUSTO, POIS ANTECIPA A CORREÇÃO DE DEFEITOS. GERALMENTE É NECESSÁRIA A UTILIZAÇÃO DE *STUBS* (MÓDULOS QUE SUBSTITUEM OUTROS MÓDULOS SUBORDINADOS) E *DRIVERS* (UM MÓDULO QUE SUBSTITUI OUTRO MÓDULO QUE SEJA RESPONSÁVEL POR CONTROLAR A CHAMADA DE UM SISTEMA), PARA SEREM UTILIZADOS NO LUGAR DOS SOFTWARES QUE ESTEJAM EVENTUALMENTE FALTANDO E PARA SIMULAR A INTERFACE ENTRE OS COMPONENTES DE SOFTWARE. 181

• **TESTE DE INTEGRAÇÃO:** NESTA ETAPA, AS UNIDADES QUE FORAM TESTADAS INDIVIDUALMENTE NO ESTÁGIO ANTERIOR SÃO TESTADAS DE FORMA INTEGRADA, BEM COMO AS INTERFACES ENTRE OS COMPONENTES. A INTEGRAÇÃO DEVE SER REALIZADA ADICIONANDO-SE OS COMPONENTES UM POR UM, E APÓS CADA PASSO UM TESTE É NECESSÁRIO (TESTE INCREMENTAL). ESTA TÉCNICA TEM A VANTAGEM DE ADIANTAR A DETECÇÃO DE DEFEITOS NO PROCESSO DE TESTES E CORRIGI-LOS MAIS RAPIDAMENTE, ENQUANTO É MAIS FÁCIL DETERMINAR AS CAUSAS DOS ERROS. POR OUTRO LADO, TEM A DESVANTAGEM DE SER UMA PRÁTICA BASTANTE CUSTOSA. SENDO ASSIM, A INTEGRAÇÃO PODE SER FEITA BASICAMENTE DE DUAS FORMAS: *TOP-DOWN* OU *BOTTOM-UP*. NA PRIMEIRA, OS TESTES SÃO REALIZADOS DE CIMA PARA BAIXO (COMEÇANDO DA GUI OU DO MENU PRINCIPAL); COMPONENTES OU SISTEMAS SÃO SUBSTITUÍDOS POR *STUBS*. NA SEGUNDA, OS TESTES COMEÇAM NA PARTE MAIS BÁSICA DO SISTEMA ATÉ O NÍVEL MAIS ALTO; COMPONENTES OU SISTEMAS SÃO SUBSTITUÍDOS POR *DRIVERS*. 182

• **TESTE DE SISTEMA:** NESTE NÍVEL O PROPÓSITO DO TESTE ESTÁ EM VERIFICAR O FUNCIONAMENTO DE TODO O SISTEMA, JÁ INTEGRADO, E ANALISAR SE ELE ESTÁ DE ACORDO COM OS REQUISITOS QUE FORAM ESPECIFICADOS. NESTE MOMENTO, NÃO SÓ SÃO REALIZADOS OS TESTES DE INTEGRAÇÃO DOS COMPONENTES DO SOFTWARE ENTRE SI, COMO TAMBÉM DESTES COMPONENTES COM UM AMBIENTE DE TESTE CORRESPONDENTE À PRODUÇÃO FINAL (HARDWARE, SOFTWARE, OUTROS SISTEMAS), DE MODO A MINIMIZAR O RISCO DE QUE FALHAS RELACIONADAS COM O AMBIENTE OPERACIONAL DO PRODUTO NÃO SEJAM ENCONTRADAS. GERALMENTE A ESTRATÉGIA DE CAIXA PRETA É UTILIZADA NESTE ESTÁGIO, MAS TESTES DE CAIXA BRANCA TAMBÉM PODEM SER REALIZADOS. 182

• **TESTE DE ACEITAÇÃO:** O TESTE DE ACEITAÇÃO CORRESPONDE AO TESTE REALIZADO PELO USUÁRIO DE FATO DO SISTEMA, NO MOMENTO EM QUE TODOS OU QUASE TODOS OS DEFEITOS ENCONTRADOS NAS ETAPAS ANTERIORES JÁ TENHAM SIDO CORRIGIDOS. O PROPÓSITO DESTE TESTE É ESTABELECEER A CONFIANÇA DO SISTEMA; ELE ESTÁ MAIS RELACIONADO COM A VALIDAÇÃO DO SISTEMA, EM QUE ESTÁ SE TENTANDO DETERMINAR SE O SISTEMA ESTÁ DE ACORDO COM OS REQUISITOS ESPECIFICADOS. NORMALMENTE OS TESTES DE ACEITAÇÃO PODEM SER DE DUAS CATEGORIAS: TESTES *ALFA* E TESTES *BETA*. OS PRIMEIROS SÃO REALIZADOS NAS INSTALAÇÕES DO DESENVOLVEDOR, QUE FICA OBSERVANDO OS USUÁRIOS UTILIZAREM O SISTEMA, E ANOTAM OS PROBLEMAS IDENTIFICADOS. JÁ OS TESTES *BETA* SÃO REALIZADOS NO AMBIENTE REAL DE TRABALHO DO USUÁRIO, QUE INSTALA O SISTEMA E TESTA, SEM A PRESENÇA DO DESENVOLVEDOR. EM SEGUIDA, UM DOCUMENTO CONTENDO OS REGISTROS DOS PROBLEMAS ENCONTRADOS É ENVIADO À ORGANIZAÇÃO DESENVOLVEDORA. 182

10.2.3. TIPOS DE TESTES

CADA TIPO DE TESTE É FOCADO EM UM GRUPO DE ATIVIDADES COM UM DETERMINADO OBJETIVO. É NECESSÁRIO PENSAR EM DIFERENTES TIPOS DE TESTES UMA VEZ QUE TESTAR A FUNCIONALIDADE DE UM COMPONENTE OU SISTEMA PODE NÃO SER SUFICIENTE EM CADA UM DOS ESTÁGIOS ENVOLVIDOS PARA SE CHEGAR AOS OBJETIVOS DOS TESTES. UM TIPO DE TESTE É FOCADO NUM OBJETIVO PARTICULAR DE TESTE, QUE PODERIA SER UM TESTE DE UMA FUNÇÃO A SER EXECUTADA PELO COMPONENTE OU SISTEMA; ALGUMA CARACTERÍSTICA NÃO FUNCIONAL; A ESTRUTURA OU ARQUITETURA DO COMPONENTE OU SISTEMA, ETC. EXISTEM VÁRIOS TIPOS DE TESTES, DEPENDENDO DO OBJETIVO DE CADA PROJETO E DE CADA ORGANIZAÇÃO. ABAIXO SERÃO APRESENTADOS ALGUNS DOS MAIS COMUNS [GRAHAM ET. AL 2007]. 182

• TESTE FUNCIONAL: ESTE TIPO DE TESTE ESTÁ FOCADO NAS REGRAS DE NEGÓCIO DO SISTEMA, OU SEJA, O FLUXO DE TRABALHO DO PROGRAMA É AVALIADO. 183

• TESTE DE RECUPERAÇÃO DE FALHA: O SISTEMA É FORCADO A FALHAR DE DIVERSAS MANEIRAS DE MODO A VERIFICAR SEU COMPORTAMENTO DIANTE DESTAS FALHAS, E REPARAR DE QUE FORMAS ELE SE RECUPERA. 183

• TESTE DE INTEROPERABILIDADE: TESTA UM PRODUTO DE SOFTWARE DE MODO A DETERMINAR SUA CAPACIDADE DE INTERAGIR COM UM OU MAIS COMPONENTES OU SISTEMAS. 183

• TESTE DE SEGURANÇA: VERIFICA SE O SISTEMA POSSUI ATRIBUTOS PARA PREVENIR ACESSOS NÃO AUTORIZADOS, ACIDENTAIS OU PROPOSITAIS, A PROGRAMAS E DADOS. 183

• TESTE DE CARGA: UM TIPO DE TESTE PARA MEDIR O COMPORTAMENTO DO SISTEMA QUANDO ESTE É SUBMETIDO A NÍVEIS ALTOS DE CARGA, DIFERENTE DAS CONDIÇÕES NORMAIS. É IMPORTANTE DETERMINAR O QUANTO DE CARGA O SISTEMA CONSEGUE SUPORTAR SEM FALHAR. 183

• TESTE DE PERFORMANCE: VERIFICA O RENDIMENTO DE UM SISTEMA, COMO O TEMPO DE RESPOSTA E PROCESSAMENTO, TAXA DE TRANSFERÊNCIA DE DADOS, PARA DIFERENTES CONDIÇÕES (CONFIGURAÇÕES, NUMERO DE USUÁRIOS, ETC) AS QUAIS O PROGRAMA É SUBMETIDO. 183

• TESTE DE ESTRESSE: TESTE CONDUZIDO PARA AVALIAR O COMPORTAMENTO DO SISTEMA DIANTE DE CONDIÇÕES QUE ULTRAPASSEM O LIMITE ESPECIFICADO NOS REQUISITOS. 183

• TESTE DE CONFIGURAÇÃO: TESTA O FUNCIONAMENTO DO SISTEMA EM DIFERENTES CONFIGURAÇÕES DE HARDWARE/SOFTWARE, TESTANDO COMPATIBILIDADE, CONFIGURAÇÃO DO SERVIDOR, TIPOS DE CONEXÕES COM A INTERNET, ETC. 183

• TESTE DE USABILIDADE: TESTES PARA DETERMINAR SE UM PRODUTO É FACILMENTE ENTENDÍVEL, FÁCIL DE APRENDER, FÁCIL DE OPERAR E ATRATIVO AOS USUÁRIOS, OU SEJA, SE O PRODUTO TEM UMA INTERFACE AMIGÁVEL PARA OS QUE UTILIZARÃO O SISTEMA. 183

• TESTE DE REGRESSÃO: TESTE DE REGRESSÃO É A ATIVIDADE DE TESTAR UMA NOVA VERSÃO DE UM SISTEMA PARA VALIDAR ESTA VERSÃO, DETECTANDO SE ERROS FORAM INTRODUZIDOS DEVIDO ÀS MUDANÇAS REALIZADAS NO SOFTWARE, E ENTÃO, GARANTIR A CORRETEDE DAS MODIFICAÇÕES. UMA VEZ QUE A RE-EXECUÇÃO DE TODOS OS TESTES É UMA ATIVIDADE BASTANTE CUSTOSA, UMA SELEÇÃO DE TESTES DE REGRESSÃO GERALMENTE É REALIZADA. 183

10.2.4. PROCESSO DE TESTES 183

O PROCESSO DE TESTES PODE SER DIVIDIDO BASICAMENTE EM CINCO ETAPAS: PLANEJAMENTO E CONTROLE, ANÁLISE E PROJETO, IMPLEMENTAÇÃO E EXECUÇÃO, AVALIAÇÃO DE CRITÉRIO DE SAÍDA E REPORTAGEM E ATIVIDADES DE ENCERRAMENTO DE TESTES [GRAHAM ET. AL 2007]. ESTAS ATIVIDADES SÃO LOGICAMENTE SEQUENCIAIS,

PORÉM, EM UM PROJETO ESPECÍFICO, PODEM SE SOBREPOR, SEREM EXECUTADAS EM PARALELO OU ATÉ MESMO SEREM REPETIDAS. CADA UMA DESTAS ATIVIDADES SERÁ DETALHA DOS SUB-TÓPICOS ABAIXO. 183

10.2.4.1. PLANEJAMENTO E CONTROLE 184

DURANTE O PLANEJAMENTO DE TESTES DEVE-SE TER CERTEZA DE QUE OS OBJETIVOS DOS CLIENTES E STAKEHOLDERS FORAM ENTENDIDOS DE MANEIRA CORRETA [GRAHAM ET. AL 2007]. BASEADOS NESTE ENTENDIMENTO, OS PROPÓSITOS DA ATIVIDADE DE TESTES PROPRIAMENTE DITA SÃO ESTABELECIDOS, E ASSIM, UMA ABORDAGEM E PLANO PARA OS TESTES É OBTIDA INCLUINDO ESPECIFICAÇÃO DAS ATIVIDADES DE TESTE. O PLANEJAMENTO DE TESTES APRESENTA AS SEGUINTE ATIVIDADES PRINCIPAIS: 184

• DETERMINAR O ESCOPO E RISCOS E IDENTIFICAR OS OBJETIVOS DE TESTE: SÃO DETERMINADOS OS SOFTWARES, COMPONENTES, SISTEMAS OU OUTROS PRODUTOS QUE DEVEM SER TESTADOS; OS RISCOS QUE DEVEM SER LEVADOS EM CONSIDERAÇÃO; E QUAL O PROPÓSITO DO TESTE (ENCONTRAR DEFEITOS, VERIFICAR SE ESTÁ DE ACORDO COM OS REQUISITOS OU DENTRO DOS PADRÕES DE QUALIDADE, ETC). 184

• DETERMINAR A ESTRATÉGIA DE TESTE: AQUI SERÃO ESTABELECIDAS AS TÉCNICAS QUE SERÃO UTILIZADAS, O QUE PRECISA DE FATO SER TESTADO (SELECIONAR E PRIORIZAR OS REQUISITOS) E QUE NÍVEL DE COBERTURA É NECESSÁRIO. SERÃO TAMBÉM ANALISADAS QUAIS PESSOAS PRECISARÃO SE ENVOLVER E EM QUE MOMENTO (DESENVOLVEDORES, USUÁRIOS, ETC), INCLUINDO A DEFINIÇÃO DA EQUIPE DE TESTE. 184

• DEFINIR RECURSOS: SÃO DEFINIDOS TODOS OS RECURSOS NECESSÁRIOS DURANTE O CICLO DE VIDA DE TESTES, TANTO RECURSOS MATERIAIS (PCS, SOFTWARE, FERRAMENTAS, ETC) COMO RECURSOS HUMANOS (PRINCIPAIS E DE APOIO). 184

• FAZER UM CRONOGRAMA PARA ANÁLISE E PROJETO, IMPLEMENTAÇÃO, EXECUÇÃO E AVALIAÇÃO DE TESTE: DEVERÁ SER ELABORADO UM CRONOGRAMA DE TODAS AS TAREFAS E ATIVIDADES, PARA QUE SEJA POSSÍVEL TERMINAR A FASE DE TESTES A TEMPO. 184

• ESTABELECEM OS CRITÉRIOS DE SAÍDA: CRITÉRIOS DE SAÍDA, COMO CRITÉRIO DE COBERTURA, POR EXEMPLO, DEVERÃO SER ESTABELECIDOS DE MODO A DETERMINAR QUANDO A ETAPA DE TESTES CHEGOU AO FIM. 184

APÓS PLANEJAR É NECESSÁRIA UMA MEDIDA DE CONTROLE PARA VERIFICAR SE TUDO ESTÁ SENDO DE ACORDO COM O PLANEJADO. É PRECISO COMPARAR O ANDAMENTO REAL COM O QUE FOI ESTABELECIDO NO PLANO DE TESTES, E TOMAR MEDIDAS CORRETIVAS QUANDO NECESSÁRIO. 184

10.2.4.2. ANÁLISE E PROJETO 184

ESTA É A ATIVIDADE EM QUE OS OBJETIVOS GERAIS DE TESTES SÃO TRANSFORMADOS EM CONDIÇÕES E PROJETOS DE TESTE TANGÍVEIS [GRAHAM ET. AL 2007]. O PROPÓSITO PRINCIPAL É IDENTIFICAR E DESCREVER OS CASOS DE TESTE PARA CADA VERSÃO DE TESTE, E IDENTIFICAR E ESTRUTURAR OS PROCEDIMENTOS DE TESTE, ESPECIFICANDO COMO EXECUTAR OS CASOS DE TESTE. AS PRINCIPAIS TAREFAS DESTA ETAPA PODEM SER DESTACADAS EM: 184

• REVISAR A BASE DE TESTES (COMO A ANÁLISE DE RISCO DO PRODUTO, REQUISITOS, ARQUITETURA, ESPECIFICAÇÃO DE PROJETO, E INTERFACES): A BASE DE TESTES É UTILIZADA PARA CRIAR OS TESTES. É POSSÍVEL COMEÇAR A PROJETAR OS TESTES DE CAIXA PRETA ANTES DA IMPLEMENTAÇÃO, UMA VEZ QUE A BASE DE TESTES PODE SER USADA PARA COMPREENDER O QUE O SISTEMA PRECISA FAZER. 184

• IDENTIFICAR E DESCREVER CASOS DE TESTE: UM CASO DE TESTE É UM CENÁRIO ASSOCIADO A UM REQUISITO; É UM TEXTO CONTENDO: IDENTIFICADOR, OBJETIVO, PRÉ-CONDICÕES DE EXECUÇÃO, ENTRADAS, PASSOS ESPECÍFICOS DO TESTE A SER EXECUTADO E RESULTADOS ESPERADOS E/OU PÓS-CONDICÕES DE EXECUÇÃO. UM CASO DE TESTE BEM PROJETADO TEM MUITA CHANCE DE ENCONTRAR UM ERRO AINDA NÃO CONHECIDO. 184

• ESTRUTURAR PROCEDIMENTOS DE TESTE: O PASSO A PASSO QUE DESCREVE COMO OS CASOS DE TESTE DEVEM SER EXECUTADOS. INCLUI O ESTADO INICIAL DA APLICAÇÃO; CONDIÇÕES DE FUNCIONAMENTO; COMO E QUANDO FORNECER OS DADOS DE ENTRADA E OBTER OS RESULTADOS; A FORMA DE AVALIAR ESTES RESULTADOS, DENTRE OUTROS. 185

• AVALIAR A CAPACIDADE DE TESTAR OS REQUISITOS: A ESPECIFICAÇÃO DE REQUISITOS DEVE SER COMPLETAMENTE CLARA, INFORMANDO AS CONDIÇÕES NECESSÁRIAS PARA SE DEFINIR OS TESTES. POR EXEMPLO, SE A PERFORMANCE DO SOFTWARE É ALGO CRÍTICO, DEVE SER CLARAMENTE ESPECIFICADO O TEMPO DE RESPOSTA MÍNIMO EM QUE O SISTEMA DEVE RESPONDER. 185

10.2.4.3. IMPLEMENTAÇÃO E EXECUÇÃO 185

UMA VEZ QUE OS CASOS E PROCEDIMENTOS DE TESTE FORAM ESPECIFICADOS EM ALTO NÍVEL NA ETAPA ANTERIOR, ESTE É O MOMENTO EM QUE O AMBIENTE SERÁ PREPARADO PARA QUE ELES SEJAM EXECUTADOS E COMPARADOS COM OS RESULTADOS DESEJADOS [GRAHAM ET. AL 2007]. ALÉM DISSO, É A ETAPA EM QUE OS COMPONENTES NECESSÁRIOS SÃO IMPLEMENTADOS PARA QUE OS TESTES SEJAM EXECUTADOS. AS PRINCIPAIS TAREFAS DESTAS DUAS FASES SERÃO DESTACADAS A SEGUIR. 185

• IMPLEMENTAÇÃO: 185

○ IMPLEMENTAR COMPONENTES: EFETUAR A IMPLEMENTAÇÃO DE NOVOS COMPONENTES DE APOIO NECESSÁRIOS À APLICAÇÃO DOS TESTES, OU MODIFICAÇÃO DE COMPONENTES JÁ EXISTENTES. FERRAMENTAS DE AUTOMAÇÃO PODEM SER UTILIZADAS OU OS COMPONENTES PODEM SER DESENVOLVIDOS EXPLICITAMENTE. 185

○ CRIAR SUÍTES DE TESTE: BASEADO NOS CASOS DE TESTE, UM CONJUNTO DE TESTES QUE NATURALMENTE TRABALHAM JUNTOS, FORMA UMA SUÍTE DE TESTE E SÃO UTILIZADOS PARA UMA EXECUÇÃO DE TESTE EFICIENTE. 185

○ IMPLEMENTAR E VERIFICAR O AMBIENTE: PREPARAR E VERIFICAR SE O AMBIENTE DE TESTE ESTÁ FUNCIONANDO CORRETAMENTE. 185

• EXECUÇÃO: 185

o EXECUTAR AS SUÍTES DE TESTE E CASOS DE TESTE INDIVIDUAIS, DE ACORDO COM OS PROCEDIMENTOS DE TESTE. PODE SER FEITO MANUALMENTE OU COM O AUXÍLIO DE FERRAMENTAS DE EXECUÇÃO DE TESTES. 185

o SEGUIR AS ESTRATÉGIAS DE TESTE DEFINIDAS NA ETAPA DE PLANEJAMENTO. 185

o CRIAR UM LOG COM AS SAÍDAS DA EXECUÇÃO DOS TESTES E REGISTRAR OS IDENTIFICADORES E VERSÕES DO SOFTWARE QUE ESTÁ SENDO TESTADO. 185

o FAZER A COMPARAÇÃO DOS RESULTADOS ESPERADOS E DOS RESULTADOS OBTIDOS. 185

o QUANDO HOUVER DIFERENÇAS ENTRE OS RESULTADOS ESPERADOS E OS RESULTADOS OBTIDOS, REGISTRAR OS DEFEITOS EM UM REPOSITÓRIO CENTRALIZADO. NÃO SE DEVE REGISTRÁ-LOS DE FORMA ALEATÓRIA. 185

o REALIZAÇÃO DE TESTES DE REGRESSÃO PARA CONFIRMAR QUE UMA FALHA ANTERIORMENTE REGISTRADA FOI DE FATO CONSERTADA. 185

10.2.4.4. AVALIAÇÃO DO CRITÉRIO DE SAÍDA E RELATÓRIO 185

ESTA É A FASE EM QUE SE DESEJA OBSERVAR SE JÁ FORAM EXECUTADOS TESTES SUFICIENTES PARA GARANTIR A QUALIDADE DESEJADA DO PRODUTO, SENDO ASSIM, CRITÉRIOS DE SAÍDA SÃO DEFINIDOS COM ESTA FINALIDADE [GRAHAM ET. AL 2007]. ESTES CRITÉRIOS INFORMAM SE UMA DADA ATIVIDADE DE TESTES PODE SER CONSIDERADA COMPLETA. AS PRINCIPAIS ATIVIDADES SÃO: 186

• CHECAR SE OS LOGS DE TESTES BATEM COM OS CRITÉRIOS DE SAÍDA ESPECIFICADOS NO PLANO DE TESTES: PROCURA-SE PELOS TESTES QUE TENHAM SIDO EXECUTADOS E AVALIADOS, E SE DEFEITOS FORAM ENCONTRADOS, CONSERTADOS OU RE-TESTADOS. 186

• VERIFICAR SE SERÁ NECESSÁRIA A INCLUSÃO DE MAIS TESTES OU SE OS CRITÉRIOS DE SAÍDA ESPECIFICADOS DEVEM SER MUDADOS: MAIS CASOS DE TESTES PODEM PRECISAR SER EXECUTADOS, SE POR ACASO ESTES NÃO TIVEREM SIDO TODOS EXECUTADOS CONFORME ESPERADO, OU SE FOR DETECTADO QUE A COBERTURA DE REQUISITOS NECESSÁRIA AINDA NÃO FOI ATINGIDA, OU ATÉ MESMO SE AUMENTARAM OS RISCOS DO PROJETO. 186

• ESCREVER UM RELATÓRIO DE RESUMO DE TESTES PARA OS STAKEHOLDERS: TODOS OS STAKEHOLDERS DEVEM SABER QUAIS TESTES FORAM EXECUTADOS E QUAIS OS RESULTADOS DESTES TESTES, DE MODO A PERCEBER QUE DECISÕES PRECISAM AINDA SER TOMADAS VISANDO A MELHORIA DA QUALIDADE DO SOFTWARE. 186

10.2.4.5. ATIVIDADES DE ENCERRAMENTO DE TESTE 186

A ATIVIDADE DE ENCERRAMENTO DE TESTE PODE SER DADA ATRAVÉS DE DIVERSOS FATORES, COMO POR EXEMPLO, AS INFORMAÇÕES NECESSÁRIAS DO PROCESSO DE TESTES JÁ FORAM ATINGIDAS; O PROJETO É CANCELADO; QUANDO UM MARCO PARTICULAR É

ALCANÇADO; OU QUANDO UMA VERSÃO DE MANUTENÇÃO OU ATUALIZAÇÃO ESTÁ CONCLUÍDA [GRAHAM ET. AL 2007]. AS ATIVIDADES PRINCIPAIS SÃO: 186

• CHECAR SE AS ENTREGAS QUE FORAM PROGRAMADAS FORAM DE FATO ENTREGUES E GARANTIR QUE TODOS OS PROBLEMAS REPORTADOS FORAM REALMENTE RESOLVIDOS. PARA OS QUE PERMANECERAM EM ABERTO DEVEM-SE REQUISITAR MUDANÇAS EM UMA FUTURA VERSÃO. 186

• FINALIZAR E ARQUIVAR OS ARTEFATOS PRODUZIDOS DURANTE O PROCESSO NECESSÁRIO PARA PLANEJAR, PROJETAR E EXECUTAR TESTES, COMO POR EXEMPLO, DOCUMENTAÇÃO, SCRIPTS, ENTRADAS, RESULTADOS ESPERADOS, ETC. É IMPORTANTE REUTILIZAR TUDO QUE FOR POSSÍVEL DESTES ARTEFATOS, POIS ASSIM SE CONSEGUE ECONOMIZAR TEMPO E ESFORÇO DO PROJETO. 186

• REPASSAR OS ARTEFATOS ANTERIORMENTE CITADOS PARA A EQUIPE DE MANUTENÇÃO, QUE IRÁ PROVER SUPORTE AOS USUÁRIOS DO SISTEMA E RESOLVER QUALQUER PROBLEMA ENCONTRADO DEPOIS DE SUA ENTREGA. 186

• AVALIAR COMO SE DEU O PROCESSO DE TESTES E ANALISAR AS LIÇÕES APRENDIDAS, QUE SERÃO DE GRANDE UTILIDADE PARA FUTURAS VERSÕES DOS PROJETOS. ESTE PASSO PODE PERMITIR NÃO SÓ MELHORIAS NO PROCESSO DE TESTES, COMO TAMBÉM MELHORIAS NO PROCESSO DE DESENVOLVIMENTO DO SOFTWARE COMO UM TODO. 186

10.2.5. TESTES AO LONGO DO CICLO DE VIDA DE SOFTWARE 186

AS ATIVIDADES DE TESTE NÃO SÃO ATIVIDADES QUE SÃO REALIZADAS SOZINHAS, MAS SIM EM PARALELO COM O CICLO DE VIDA DE DESENVOLVIMENTO DO SOFTWARE. DESSA FORMA, A ESCOLHA DO CICLO DE VIDA DO PROJETO IRÁ AFETAR DIRETAMENTE AS ATIVIDADES DE TESTE. O PROCESSO DE DESENVOLVIMENTO ADOTADO DEPENDE MUITO DOS OBJETIVOS E PROPÓSITOS DO PROJETO. PORTANTO, O MODO COMO AS ATIVIDADES DE TESTE SÃO ESTRUTURADAS DEVE SE AJUSTAR AO MODELO DE DESENVOLVIMENTO DE SOFTWARE, OU DO CONTRÁRIO, NÃO CONSEGUIRÁ OBTER O SUCESSO DESEJADO. 186

UM MODELO DE DESENVOLVIMENTO DE SOFTWARE BASTANTE CONHECIDO É O MODELO EM CASCATA, QUE COMO O PRÓPRIO NOME JÁ SUGERE, TEM SUA BASE VOLTADA A UM DESENVOLVIMENTO SEQUENCIAL DAS ATIVIDADES. AS PRIMEIRAS ATIVIDADES COMEÇAM NO TOPO DA CASCATA, E ENTÃO VÃO SEGUINDO SEQUENCIALMENTE ATRAVÉS DAS VÁRIAS ATIVIDADES DE CONCEPÇÃO DO PROJETO, E FINALMENTE TERMINANDO COM A ETAPA DE IMPLEMENTAÇÃO. APÓS ISSO, É QUE AS ATIVIDADES DE TESTE SÃO INTRODUZIDAS, E DESSA FORMA OS DEFEITOS SÓ PODEM SER DETECTADOS BEM PERTO DA FASE DE IMPLEMENTAÇÃO [GRAHAM ET. AL 2007]. A FIGURA 10.5 ILUSTRÁ O MODELO EM CASCATA. 187

COM O OBJETIVO DE TENTAR CONTORNAR OS PROBLEMAS DO MODELO EM CASCATA, FOI DESENVOLVIDO O MODELO V, QUE Foca NOS TESTES DO PRODUTO DURANTE TODO O CICLO DE DESENVOLVIMENTO PARA CONSEGUIR UMA DETECCÃO ADIANTADA DE DEFEITOS. A IDÉIA É QUE AS ATIVIDADES DE TESTES NÃO SÃO SIMPLEMENTE UMA FASE ÚNICA, MAS PELO CONTRÁRIO, COMO JÁ FOI VISTO NA SESSÃO ANTERIOR, SE FAZ NECESSÁRIA TODA UMA PREPARAÇÃO, PASSANDO POR ETAPAS DE PLANEJAMENTO, ANÁLISE, PROJETO, ETC, QUE DEVEM SER EXECUTADAS EM PARALELO COM AS ATIVIDADES DE DESENVOLVIMENTO. 187

TODOS OS ARTEFATOS GERADOS PELOS DESENVOLVEDORES E ANALISTAS DE NEGÓCIO DURANTE O DESENVOLVIMENTO, PROVÊM A BASE DE TESTES EM UM OU MAIS NÍVEIS, PROMOVEDO AS ATIVIDADES DE TESTE MAIS CEDO, DEFEITOS PODEM SER GERALMENTE ENCONTRADOS NOS DOCUMENTOS DA BASE DE TESTES. O MODELO V DEMONSTRA COMO AS ATIVIDADES DE VERIFICAÇÃO E VALIDAÇÃO PODEM SER EXECUTADAS EM CONJUNTO COM CADA FASE DO CICLO DE VIDA DE DESENVOLVIMENTO DO SOFTWARE. 187

10.3. INSPEÇÃO DE SOFTWARE 188

COMO EXPLICADO NA SESSÃO ANTERIOR, A INSPEÇÃO DE SOFTWARE É UMA TÉCNICA ESTÁTICA DO PROCESSO DE V & V, EM QUE SÃO EFETUADAS REVISÕES NO SISTEMA COM O OBJETIVO DE ENCONTRAR DEFEITOS E ENTÃO, CORRIGI-LOS. O OBJETIVO PRINCIPAL DAS INSPEÇÕES É GARANTIR QUE DEFEITOS SEJAM REPARADOS O MAIS CEDO POSSÍVEL NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE, UMA VEZ QUE QUANTO MAIS EVOLUIDO O SOFTWARE ESTIVER, MAIS DIFÍCIL SERÁ PARA ENCONTRAR OS ERROS E MAIS CUSTOSO AINDA CONSERTÁ-LOS. QUALQUER ARTEFATO PRODUZIDO NO DESENVOLVIMENTO DO SOFTWARE PODE SER UTILIZADO NO PROCESSO DE INSPEÇÃO, COMO REQUISITOS, MODELO DE PROJETO OU CÓDIGO. 188

O MODELO CMMI EXIGE A REALIZAÇÃO DE REVISÕES COMO UMA PRÁTICA ESPECÍFICA DO PROCESSO DE VERIFICAÇÃO, DEMONSTRANDO ASSIM SUA IMPORTÂNCIA NA GARANTIA DA QUALIDADE DO PRODUTO. SEGUNDO FAGAN, A UTILIZAÇÃO DE INSPEÇÕES INFORMAIS DE SOFTWARE CAPTURA EM TORNO DE 60% DOS ERROS EM UM PROGRAMA [FAGAN 1986]. MILLS ET AL. SUGERE QUE UMA APLICAÇÃO MAIS FORMAL DE INSPEÇÃO DE SOFTWARE PODE DETECTAR ATÉ MAIS DE 90% DOS ERROS DE UM PROGRAMA [MILLS ET AL. 1987]. SELBY E BASILI COMPARAM EMPÍRICAMENTE A EFETIVIDADE DE INSPEÇÕES E TESTES. ELES PERCEBERAM QUE A REVISÃO DE CÓDIGO ESTÁTICA SE MOSTRAVA MAIS EFETIVA E MENOS CARA DO QUE A PROCURA POR ERROS UTILIZANDO TESTES [SELBY ET AL. 1987]. 188

10.3.1. A EQUIPE DE INSPEÇÃO (PARTICIPANTES) 189

A EQUIPE DE INSPEÇÃO É COMPOSTA POR UM PEQUENO GRUPO DE PESSOAS QUE POSSUAM INTERESSE E CONHECIMENTO DO PRODUTO. GERALMENTE O TAMANHO DA EQUIPE VARIA DE QUATRO A SETE PARTICIPANTES, E O NÚMERO MÍNIMO É DE TRÊS PESSOAS. EQUIPES MAIORES SÃO NORMALMENTE UTILIZADAS PARA ANALISAR DOCUMENTOS DE MAIS ALTO NÍVEL DO PRODUTO, ENQUANTO QUE TIMES MENORES SÃO PREFERÍVEIS AO SE INSPECIONAR DETALHES MAIS TÉCNICOS. 189

É BASTANTE INTERESSANTE PARA O PROCESSO DE INSPEÇÃO QUE EXISTA UMA BOA VARIEDADE DE INSPETORES, PERTENCENTES A DIFERENTES ÁREAS DE CONHECIMENTO. O PAPEL DE CADA PARTICIPANTE SERÁ EXPLICADO ABAIXO. 189

- AUTOR: É O CRIADOR (DESENVOLVEDOR) DO ARTEFATO QUE SERÁ INSPECIONADO. SUAS PRINCIPAIS RESPONSABILIDADES SÃO: CORRIGIR OS PROBLEMAS DETECTADOS DURANTE O PROCESSO DE INSPEÇÃO, PROVER UMA VISÃO GERAL DO PRODUTO AOS DEMAIS PARTICIPANTES E TIRAR QUAISQUER DÚVIDAS QUE SURGIREM COM RELAÇÃO AO ARTEFATO DESENVOLVIDO. 189

- INSPETOR: EXAMINA O PRODUTO ANTES E DURANTE A REUNIÃO DE INSPEÇÃO (FASE DE PREPARAÇÃO) DE MODO A TENTAR ENCONTRAR DEFEITOS. PODE TAMBÉM IDENTIFICAR

PROBLEMAS AMPLOS QUE ESTÃO FORA DO ESCOPO DA EQUIPE DE INSPEÇÃO, COMO TAMBÉM SUGERIR MELHORIAS. 189

• **LEITOR: PESSOA RESPONSÁVEL POR APRESENTAR O ARTEFATO AOS DEMAIS PARTICIPANTES DO PROCESSO DE INSPEÇÃO DURANTE A REUNIÃO. UMA PESSOA QUE USARÁ O PRODUTO NUMA PRÓXIMA ETAPA DO SEU CICLO DE VIDA É UM CANDIDATO FORTE PARE ESTA TAREFA, UMA VEZ QUE A ATIVIDADE DE LER SOBRE O PRODUTO IRÁ PERMITIR A ESTE POTENCIAL USUÁRIO SE TORNAR BASTANTE FAMILIAR COM O PRODUTO.**

189

• **ESCRITOR: TEM O PAPEL DE REGISTRAR AS INFORMAÇÕES SOBRE CADA DEFEITO ENCONTRADO DURANTE A REUNIÃO, QUE INCLUEM: A LOCALIZAÇÃO DO DEFEITO, UM RESUMO DO PROBLEMA, SUA CLASSIFICAÇÃO E UMA IDENTIFICAÇÃO DO INSPECTOR QUE O ENCONTROU. TODAS AS DECISÕES E RECOMENDAÇÕES FEITAS TAMBÉM SÃO REGISTRADAS.**

189

• **MODERADOR: O MODERADOR TEM O PAPEL MAIS CRÍTICO NO PROCESSO DE INSPEÇÃO E POR ESTE MOTIVO FAZ-SE NECESSÁRIO UM TREINAMENTO MAIS APROFUNDADO DO QUE OS OUTROS MEMBROS DA EQUIPE. ELE É A PESSOA QUE LIDERA TODA A EQUIPE E PARTICIPA ATIVAMENTE DE TODAS AS ETAPAS. DENTRE SUAS PRINCIPAIS RESPONSABILIDADES PODEMOS DESTACAR: SELECIONAR E LIDERAR A EQUIPE DE INSPEÇÃO, DISTRIBUIR O MATERIAL A SER INSPECIONADO, AGENDAR AS REUNIÕES, ATUAR COMO MODERADOR NOS ENCONTROS, SUPERVISIONAR A CORREÇÃO DOS DEFEITOS, E EMITIR RELATÓRIO DE INSPEÇÃO. UMA OUTRA RESPONSABILIDADE MUITO IMPORTANTE DO MODERADOR É GARANTIR QUE O FOCO DA REUNIÃO SE MANTENHA EM ENCONTRAR FALHAS NO PRODUTO, E NÃO EM ACUSAR O AUTOR DOS PROBLEMAS ENCONTRADOS.**

189

10.3.2. O PROCESSO DE INSPEÇÃO DE SOFTWARE (ETAPAS) 189

O PROCESSO TRADICIONAL DE INSPEÇÃO DE SOFTWARE [FAGAN 1976] É DEFINIDO POR SEIS ESTÁGIOS, CADA UM REPRESENTADO POR SEU PRINCIPAL RESPONSÁVEL. A FIGURA 10.7 ILUSTRA ESTA SEQUÊNCIA DE ETAPAS E EM SEGUIDA CADA UMA DAS ETAPAS SERÁ EXPLICADA DETALHADAMENTE.

189

• **PLANEJAMENTO: O MODERADOR É A PESSOA RESPONSÁVEL POR ESTA ETAPA. O PLANEJAMENTO ENVOLVE SELECIONAR A EQUIPE, CHECAR SE O PRODUTO ESTÁ PRONTO PARA INSPEÇÃO, ORGANIZAR A REUNIÃO, DELEGAR AS ATIVIDADES DE CADA MEMBRO E GARANTIR A COMPLETUDE DOS MATERIAIS A SEREM INSPECIONADOS. NESTA ETAPA O MODERADOR TAMBÉM DEVE VERIFICAR SE O MATERIAL A SER INSPECIONADO POSSUI UM TAMANHO ADEQUADO PARA UMA ÚNICA REUNIÃO. CASO CONTRÁRIO, O MATERIAL DEVERÁ SER DIVIDIDO EM TAMANHOS MENORES, COM INSPECÇÕES A SEREM REALIZADAS PARA CADA UMA DESTAS PARTES.**

190

• **VISÃO GERAL: NESTA ETAPA O AUTOR APRESENTA O PRODUTO AOS DEMAIS MEMBROS DA EQUIPE, DESCREVENDO O QUE O PROGRAMA É SUPOSTO FAZER. O MODERADOR É RESPONSÁVEL POR DECIDIR SE ESTA ETAPA SE FAZ REALMENTE NECESSÁRIA, POIS SE A EQUIPE JÁ FOR BEM FAMILIARIZADA COM O MATERIAL A SER INSPECIONADO OU NOVAS TÉCNICAS NÃO ESTEJAM SENDO APLICADAS, ESTE ESTÁGIO É DISPENSÁVEL.**

190

• **PREPARAÇÃO: ESTE É O MOMENTO EM QUE CADA MEMBRO DO TIME DE INSPEÇÃO ESTUDA INDIVIDUALMENTE A ESPECIFICAÇÃO E O PROGRAMA A SER INSPECIONADO, E**

PROCURA POR DEFEITOS NO MATERIAL. TODOS OS POSSÍVEIS DEFEITOS DEVEM SER REGISTRADOS NUM LOG DE PREPARAÇÃO. ASSIM COMO O TEMPO QUE FOI GASTO NA PREPARAÇÃO. O MODERADOR É ENCARREGADO DE ANALISAR OS LOGS ANTES DA REUNIÃO DE INSPEÇÃO PARA DETERMINAR SE A EQUIPE ESTÁ PREPARADA PARA SUAS TAREFAS, E CASO CONTRÁRIO, ELE PODE REMARCAR A REUNIÃO. 190

• REUNIÃO: NESTA ETAPA, O PASSO A PASSO PRINCIPAL CONSISTE NA LEITURA E INTERPRETAÇÃO DO PRODUTO, PELO LEITOR; EM SEGUIDA O AUTOR TIRA QUAISQUER DÚVIDAS QUE EVENTUALMENTE SURGIREM COM RELAÇÃO AO MATERIAL, E A EQUIPE DE INSPETORES ENTÃO IDENTIFICAM OS POSSÍVEIS DEFEITOS. ESTA REUNIÃO DEVE SER CURTA, NÃO PODENDO PASSAR MAIS DO QUE DUAS HORAS, E DEVE SER FOCADA NA DETECÇÃO DE DEFEITOS, CONFORMIDADE COM O PADRÃO E PROGRAMAÇÃO DE MÁ QUALIDADE. O TIME DE INSPEÇÃO NÃO DEVE DISCUTIR COMO ESTES DEFEITOS PODERIAM SER CORRIGIDOS E NEM SUGERIR MUDANÇAS EM OUTROS COMPONENTES. 190

• RE-TRABALHO: O PROPÓSITO DO RE-TRABALHO É CORRIGIR OS DEFEITOS IDENTIFICADOS DURANTE A REUNIÃO DE INSPEÇÃO. O AUTOR É A PESSOA RESPONSÁVEL POR ESSAS CORRECÇÕES, DEVENDO CORRIGIR EM PRIMEIRO LUGAR OS DEFEITOS CONSIDERADOS MAIS RELEVANTES E GRAVES, E CORRIGINDO OS DE MENOR IMPORTÂNCIA APENAS SE O TEMPO PERMITIR. 191

• ACOMPANHAMENTO: AQUI O MODERADOR DEVE DECIDIR SE UMA NOVA INSPEÇÃO É NECESSÁRIA OU NÃO. ELE DEVE ANALISAR O MATERIAL CORRIGIDO PELOS AUTORES E VERIFICAR SE OS DEFEITOS FORAM CORRIGIDOS COM SUCESSO. O MODERADOR PODE INCLUIR REVISORES ADICIONAIS NESTA ETAPA SE FOREM NECESSÁRIOS CONHECIMENTOS TÉCNICOS EXTRAS. SE TODOS OS PROBLEMAS MAIS RELEVANTES FOREM RESOLVIDOS, OS PROBLEMAS EM ABERTO SOLUCIONADOS, E O PRODUTO SATISFIZER AOS CRITÉRIOS DE SAÍDA, O MODERADOR APROVA O *RELEASE* DO PRODUTO. SE AS CONDIÇÕES NÃO FORAM ATINGIDAS, AINDA SERÁ NECESSÁRIO MAIS TEMPO NA ETAPA DE RE-TRABALHO. 191

10.3.3. FERRAMENTAS DE APOIO AO PROCESSO DE INSPEÇÃO 191

BASEADO NA CLASSIFICAÇÃO DE *GROUPWARE* (SOFTWARES VOLTADOS PARA O APOIO A ATIVIDADES DE TRABALHO EM GRUPO) E NAS CONSTANTES MUDANÇAS TECNOLÓGICAS, [HEDBERG 2004] IDENTIFICOU QUATRO GERAÇÕES DE FERRAMENTAS DE INSPEÇÃO DE SOFTWARE: 191

1. PRIMEIRAS FERRAMENTAS (*EARLY TOOLS*) 191

2. FERRAMENTAS DISTRIBUÍDAS (*DISTRIBUTED TOOLS*) 191

3. FERRAMENTAS ASSÍNCRONAS (*ASYNCHRONOUS TOOLS*) 191

4. FERRAMENTAS BASEADAS EM WEB (*WEB-BSED TOOLS*) 191

NOTA-SE QUE AS PRIMEIRAS FERRAMENTAS A SURGIREM FORAM CLASSIFICADAS COMO PRIMEIRAS FERRAMENTAS, NO INÍCIO DA DÉCADA DE 90 E LOGO EM SEGUIDA VIERAM AS FERRAMENTAS DISTRIBUÍDAS. NO FINAL DA DÉCADA DE 90 SURGIRAM AS FERRAMENTAS PARA INTERNET. 191

AS FERRAMENTAS DA PRIMEIRA GERAÇÃO SÃO AQUELAS QUE APENAS PERMITEM O TRABALHO DE TODA A EQUIPE NO MESMO AMBIENTE E AO MESMO TEMPO (INSPECÇÕES SÍNCRONAS). A SEGUNDA JÁ PERMITE QUE A EQUIPE POSSA TRABALHAR DE FORMA DISTRIBUÍDA, OU SEJA, EM LUGARES DIFERENTES, PORÉM AINDA É PRECISO QUE SEJA AO MESMO TEMPO (INSPECÇÕES DISTRIBUÍDAS). A TOTAL INDEPENDÊNCIA DE TEMPO E LUGAR FOI INTRODUZIDA NA TERCEIRA GERAÇÃO, COM AS FERRAMENTAS ASSÍNCRONAS. AS FERRAMENTAS DA QUARTA GERAÇÃO TAMBÉM SÃO ASSÍNCRONAS, DIFERENCIANDO-SE DAS DEMAIS DEVIDO A SUA BASE TECNOLÓGICA.

191

A SEGUIR SERÁ APRESENTADA UMA FERRAMENTA REPRESENTANTE DE CADA GERAÇÃO INTRODUZIDA ANTERIORMENTE [WONG 2006]. A FERRAMENTA ICICLE REPRESENTARÁ A GERAÇÃO DE PRIMEIRAS FERRAMENTAS. EM SEGUIDA A FERRAMENTA SCRUTINY EXEMPLIFICARÁ AS FERRAMENTAS DISTRIBUÍDAS. ASSIST ILUSTRARÁ AS FERRAMENTAS ASSÍNCRONAS, E FINALMENTE, IBIS SERÁ A REPRESENTANTE DAS FERRAMENTAS BASEADAS EM WEB.

191

• ICICLE – O ICICLE (INTELLIGENT CODE INSPECTION ENVIRONMENT IN A C LANGUAGE ENVIRONMENT) É O PRIMEIRO SOFTWARE DE REVISÃO PUBLICADO E VISA APOIAR O PROCESSO TRADICIONAL DE INSPEÇÃO DE SOFTWARE. COMO O PRÓPRIO NOME JÁ SUGERE, ELE FOI DESENVOLVIDO PARA O CONTEXTO ESPECÍFICO DE INSPEÇÃO DE CÓDIGO C E C++, PODENDO SER USADO PARA O AUXÍLIO DA INSPEÇÃO DO CÓDIGO, TANTO NAS FASES DE PREPARAÇÃO INDIVIDUAL COMO NAS REUNIÕES EM GRUPO. A REUNIÃO DE INSPEÇÃO EM GRUPO DEVE SER REALIZADA NO MESMO LOCAL E A INSPEÇÃO INDIVIDUAL PERMITE ENTRAR COM COMENTÁRIOS EM CADA LINHA DE CÓDIGO. A FERRAMENTA NÃO SE APLICA A INSPECÇÕES MAIS GENÉRICAS, LIMITANDO O TIPO DE ARTEFATO A SER INSPECIONADO E A TÉCNICA DE DETECÇÃO DE DEFEITOS, MAS PODE, ENTRETANTO, SER UTILIZADA PARA INSPECIONAR LINHAS DE TEXTO NUMA ANÁLISE INICIAL. UM DOS PRINCIPAIS OBJETIVOS DESTA FERRAMENTA É O DE AJUDAR OS INSPETORES DE CÓDIGO A ENCONTRAREM DEFEITOS ÓBVIOS.

191

• SCRUTINY – O SCRUTINY É UMA FERRAMENTA COLABORATIVA ONLINE, SENDO A PRIMEIRA A PERMITIR QUE OS MEMBROS DO TIME DE INSPEÇÃO SE ENCONTRASSEM DISPERSOS GEOGRAFICAMENTE, PODENDO SER USADA TANTO DE FORMA SÍNCRONA COMO ASSÍNCRONA. ELA PODE SER INTEGRADA COM OUTRAS FERRAMENTAS E CUSTOMIZADA PARA APOIAR DIFERENTES PROCESSOS DE DESENVOLVIMENTO. ATUALMENTE APENAS SUPORTA INSPECÇÕES DE TEXTOS. A FERRAMENTA É BASEADA NUM PROCESSO DE INSPEÇÃO DIVIDIDO EM QUATRO ETAPAS. NO PRIMEIRO ESTÁGIO, DE INICIAÇÃO, O MODERADOR DISPONIBILIZA O DOCUMENTO A SER INSPECIONADO NA FERRAMENTA. NO PRÓXIMO ESTÁGIO, PREPARAÇÃO, OS INSPETORES INSEREM SEUS COMENTÁRIOS A SEREM DISCUTIDOS NA REUNIÃO. DEPOIS, NA FASE DE RESOLUÇÃO, O MODERADOR GUIA OS INSPETORES ATRAVÉS DOS DOCUMENTOS E DOS DEFEITOS COLETADOS. FINALMENTE, NO ESTÁGIO DE FINALIZAÇÃO, APÓS AS DISCUSSÕES E ACORDOS REFERENTES AOS DEFEITOS LEVANTADOS, A FERRAMENTA FORNECE UM RESUMO DOS DEFEITOS QUE FORAM DISCUTIDOS.

192

• ASSIST – ASYNCHRONOUS/ SYNCHRONOUS SOFTWARE INSPECTION SUPPORT TOOL FOI DESENVOLVIDA PARA PROVER INSPECÇÕES INDIVIDUAIS E EM GRUPO. COMO O NOME SUGERE, PERMITE INSPECÇÕES SÍNCRONAS E ASSÍNCRONAS, COM REUNIÕES TANTO EM LOCAIS DIFERENTES COMO NO MESMO AMBIENTE. UTILIZA UMA LINGUAGEM DE DEFINIÇÃO DE PROCESSO DE INSPEÇÃO (IPDL) E UM SISTEMA FLEXÍVEL PARA O TIPO DE DOCUMENTO INSPECIONADO, PERMITINDO O SUPORTE A QUALQUER TIPO DE PROCESSO DE INSPEÇÃO DE SOFTWARE. INSPEÇÃO DE CÓDIGO, COLETAS DE DADOS PARA MÉTRICAS E CÁLCULOS PARA APOIO AS INSPECÇÕES TAMBÉM ESTÃO PRESENTES NESTA FERRAMENTA. É

BASEADA NUMA ARQUITETURA CLIENTE/SERVIDOR, EM QUE O SERVIDOR É USADO COMO UM REPOSITÓRIO CENTRAL DE DOCUMENTOS E OUTROS TIPOS DE DADOS. UM BROWSER C++ PODE AUTOMATICAMENTE APRESENTAR ITENS RELEVANTES DE CHECKLIST PARA A SESSÃO DE CÓDIGO INSPECIONADO.

192

• IBIS – INTERNET-BASED INSPECTION SYSTEM É UMA FERRAMENTA BASEADA EM WEB COM NOTIFICAÇÕES POR EMAIL QUE AUXILIA NO PROCESSO DE INSPEÇÃO DESENVOLVIDO POR FAGAN. PERMITE QUE AS INSPECÇÕES SEJAM REALIZADAS ENTRE PESSOAS GEOGRAFICAMENTE DISTRIBUÍDAS E POSSUI UMA INTERFACE BASTANTE LEVE E AMIGÁVEL, TENDO TODA SUA ESTRUTURA E DADOS ARMAZENADOS EM ARQUIVOS XML. ELA NÃO LIMITA O TIPO DE ARTEFATO A SER INSPECIONADO E PROVÊ SUPORTE A DECISÕES, APOIO A ANOTAÇÕES E CHECKLISTS. AS PRINCIPAIS VANTAGENS DESTA FERRAMENTA SÃO: PERMITE QUE OS INSPETORES ACESSEM A APLICAÇÃO DE SEUS PRÓPRIOS COMPUTADORES; ADMITE QUE A INSPEÇÃO SEJA REALIZADA COM INTEGRANTES DA EQUIPE DISTRIBUÍDOS EM LOCAIS DIFERENTES, ATÉ MESMO EM PAÍSES DIFERENTES; PERMITE QUE ESPECIALISTAS DIFERENTES PARTICIPEM DA REUNIÃO, PODENDO SER ESPECIALISTAS DE OUTRO DEPARTAMENTO OU MESMO FORA NA ORGANIZAÇÃO.

192

10.4. MODELOS DE MATURIDADE DE TESTES DE SOFTWARE

193

PARA SE CONSTRUIR SOFTWARE COM QUALIDADE, É NECESSÁRIO QUE SE TENHA UM PROCESSO DE TESTES BEM DEFINIDO E QUE ELE ESTEJA ALINHADO AO PROCESSO DE DESENVOLVIMENTO. NESTA SEÇÃO SERÃO VISTOS TRÊS MODELOS DE MATURIDADE DE TESTE DE SOFTWARE, OS QUAIS INDICAM COMO CRIAR E/OU MELHORAR O PROCESSO DE TESTES.

193

10.4.1. PROCESSO DE MELHORIA DE TESTES – TPI

193

10.4.1.1. ESCOPO DO TPI

193

10.4.1.2. ÁREAS CHAVE

194

10.4.1.3. PASSOS PARA IMPLANTAR A MELHORIA

195

10.4.2. TMM – TEST MATURITY MODEL

196

10.4.2.1. NÍVEIS DE MATURIDADE DO TMM

197

10.4.3. TIM – TEST IMPROVEMENT MODEL

198

10.4.3.1. MODELO DE MATURIDADE

198

10.4.3.2. ÁREAS CHAVE

199

CONSIDERAÇÕES FINAIS

202

O DESENVOLVIMENTO DE SOFTWARE ENGOBA UM MERCADO DE EXTREMA COMPETITIVIDADE. TENDO EM VISTA QUE OS SISTEMAS QUE APRESENTAM MELHOR QUALIDADE GARANTEM SEU ESPAÇO NO MERCADO, AS EMPRESAS QUE OS DESENVOLVEM TÊM INVESTIDO BASTANTE PARA ASSEGURAR A QUALIDADE DE SEUS PRODUTOS E GARANTIR A SATISFAÇÃO DOS CLIENTES. A QUALIDADE DE UM PRODUTO PODE SER DEFINIDA COMO SUA CAPACIDADE DE CUMPRIR OS REQUISITOS INICIALMENTE ESTIPULADOS PELOS CLIENTES, E SENDO ASSIM, ESTÁ DIRETAMENTE RELACIONADA À QUALIDADE DO PROCESSO DE DESENVOLVIMENTO. POR ESTE MOTIVO, TEM SURGIDO UMA GRANDE DEMANDA AO INCENTIVO DE PESQUISAS QUE LEVEM EM CONSIDERAÇÃO A PROCURA POR FORMAS DE MELHORIA DA QUALIDADE DOS PRODUTOS. 202

ESTE CAPÍTULO PROCUROU INTRODUIR AO LEITOR BOAS PRÁTICAS NO QUE DIZ RESPEITO À QUALIDADE DOS PRODUTOS, APRESENTANDO UM CONJUNTO DE NORMAS QUE REPRESENTAM A PADRONIZAÇÃO MUNDIAL PARA AVALIAÇÃO DA QUALIDADE DE PRODUTOS DE SOFTWARE. AS ATIVIDADES DE TESTE E INSPEÇÃO TAMBÉM FORAM DESTACADAS COMO FORMA DE ENCONTRAR DEFEITOS NO SOFTWARE E CORRIGI-LOS, ANTES DE ENTREGAR O PRODUTO A SEUS CLIENTES, E ANALISAR SE O SISTEMA FAZ O QUE É SUPOSTO FAZER. FINALMENTE, MODELOS DE MATURIDADE DE TESTES FORAM APRESENTADOS COMO MAIS UMA TENTATIVA DE ALCANÇAR MELHORIAS NA QUALIDADE DO PROCESSO DE TESTE DE SOFTWARE, QUE AFETA DIRETAMENTE A QUALIDADE DO PRODUTO. 202

EXERCÍCIOS 203

1. QUAIS SÃO AS DIRETRIZES PARA USO DA NORMA NBR ISO/IEC 9126-1? 203

2. A QUE SE PROPÕE A NORMA ISO 12119? 203

3. QUE SUBDIVISÕES DA NORMA ISO 14598 ESTABELECEM ITENS NECESSÁRIOS PARA O SUPORTE À AVALIAÇÃO? 203

4. QUAIS SÃO OS COMPONENTES DO PROJETO SQUARE? DEFINA-OS. 203

5. QUAL A DIFERENÇA ENTRE TESTES E INSPEÇÕES DE SOFTWARE? 203

6. CITE 5 TIPOS DE TESTES E EXPLIQUE CADA UM DELES. 203

7. QUAIS OS ESTÁGIOS DE TESTES POSSÍVEIS E QUAIS AS CARACTERÍSTICAS DE CADA UM DELES? 203

8. O QUE SÃO TESTES BETA? 203

9. O QUE SÃO TESTES DE REGRESSÃO? 203

10. QUAL A DIFERENÇA ENTRA A ABORDAGEM DE CAIXA PRETA E A ABORDAGEM DE CAIXA BRANCA? 203

11. QUAIS SÃO OS PAPÉIS EXISTENTES NA EQUIPE DE INSPEÇÃO DE SOFTWARE E QUAIS SUAS RESPONSABILIDADES? 203

<u>12. QUAIS SÃO AS ETAPAS DO PROCESSO DE INSPEÇÃO DE SOFTWARE? EXPLIQUE CADA UMA DELAS.</u>	203
<u>13. EXPLIQUE COMO É FEITA A IMPLANTAÇÃO DA MELHORIA NO TPI.</u>	203
<u>14. DEFINA OS NÍVEIS DE MATURIDADE DO TMM.</u>	203
<u>15. NO ASPECTO ORGANIZAÇÃO, COMO SÃO CARACTERIZADOS OS NÍVEIS DE MATURIDADE DO TIM?</u>	203
SUGESTÕES DE LEITURA	204
<u>PARA CONHECER MAIS SOBRE NORMAS DE QUALIDADE DE PRODUTO DE SOFTWARE, É RECOMENDADA A LEITURA DO LIVRO TECNOLOGIA DA INFORMAÇÃO: QUALIDADE DE PRODUTO DE SOFTWARE, GUERRA & COLOMBO 2009.</u>	204
<u>PARA AMPLIAR O ENTENDIMENTO SOBRE O ASSUNTO DE TESTE DE SOFTWARE É RECOMENDADA A LEITURA DO LIVRO <i>FOUNDATIONS OF SOFTWARE TESTING</i>, GRAHAM, D., VEENENDAAL, E. V., EVANS, I. AND BLACK, R., 2007. ESTE LIVRO É UTILIZADO POR PESSOAS QUE DESEJAM TIRAR O CERTIFICADO ISTQB (INTERNATIONAL SOFTWARE TESTING QUALIFICATIONS BOARD), PORTANTO, É MUITO INTERESSANTE PARA ADQUIRIR MELHORES CONHECIMENTOS SOBRE ESTE CONTEÚDO.</u>	204
<u>PARA UM MELHOR CONHECIMENTO SOBRE OS CONCEITOS E O PROCESSO DE INSPEÇÃO DE SOFTWARE É SUGERIDA A LEITURA DE <i>DESIGN AND CODE INSPECTION TO REDUCE ERRORS IN PROGRAM DEVELOPMENT</i>, FAGAN, M.E.,1976.</u>	204
<u>PARA SE APROFUNDAR MAIS SOBRE AS FERRAMENTAS DE INSPEÇÃO DE SOFTWARE É RECOMENDADA A LEITURA DE <i>MODERN SOFTWARE REVIEW TECHNIQUES AND TECHNOLOGIES</i>, WONG, Y. K., 2006.</u>	204
<u>PARA MELHOR CONHECIMENTO SOBRE O TPI (TEST PROCESS IMPROVEMENT) É RECOMENDADA A LEITURA DO LIVRO <i>TEST PROCESS IMPROVEMENT A PRACTICAL STEP-BY-STEP GUIDE TO STRUCTURED TESTING</i>, KOOMEN & POL, 1999.</u>	204
<u>PARA APROFUNDAR A LEITURA SOBRE TMM (TEST MATURITY MODEL), É SUGERIDA A LEITURA DO LIVRO <i>A MODEL TO ASSESS TESTING PROCESS MATURITY</i>, BURNSTEIN & GROM, 1998. TÓPICOS DE PESQUISA</u>	204
TÓPICOS DE PESQUISA	205
<u>EXISTEM VÁRIOS ESTUDOS ATUALMENTE NA ACADEMIA NO QUE DIZ RESPEITO À SELEÇÃO DE TESTES DE REGRESSÃO, UMA VEZ QUE EXECUTAR TODOS OS CASOS DE TESTE NOVAMENTE SEMPRE QUE UMA NOVA VERSÃO DO SISTEMA FOR LIBERADA É UMA PRÁTICA INVIÁVEL. DESSA FORMA, VÁRIAS PESQUISAS E PROPOSTAS DE SOLUÇÕES E TÉCNICAS PARA REALIZAR UMA QUANTIDADE SUFICIENTE DE TESTES QUE ATINJA A COBERTURA NECESSÁRIA PARA GARANTIR A CORRETEDE DO SOFTWARE PODEM SER ENCONTRADAS NA LITERATURA.</u>	205

OUTRA ÁREA DE PESQUISA BASTANTE DESAFIADORA NA ÁREA DE TESTE DE SOFTWARE É A GERAÇÃO AUTOMÁTICA DE CASOS DE TESTE, CONSIDERANDO QUE A ELABORAÇÃO DE CASOS TESTES MANUALMENTE É UM PROCESSO QUE CONSUME MUITO TEMPO E ESFORÇO. SENDO ASSIM, DIVERSAS PROPOSTAS SÃO ELABORADAS DIA APÓS DIA COM O OBJETIVO DE TORNAR O PROCESSO DE TESTE MAIS ÁGIL, MENOS SUSCEPTÍVEL A ERROS E DEPENDENTE DA INTERAÇÃO HUMANA. 205

NA ÁREA DE INSPEÇÃO DE SOFTWARE, GRANDES DESAFIOS PODEM SER OBSERVADOS COM O OBJETIVO DE ENCONTRAR ESTRATÉGIAS PARA DIMINUIR A QUANTIDADE DE DEFEITOS DE UM SOFTWARE. NA LITERATURA, PODEM SER ENCONTRADAS PESQUISAS E ARTIGOS COM ESTUDOS FOCADOS NESTE OBJETIVO. 205

NA ÁREA DE MODELO DE MATURIDADE DE TESTES, HÁ UMA ORGANIZAÇÃO, CHAMADA *TMMI FOUNDATION*, SEM FINS LUCRATIVOS, EM DUBLIN – IRLANDA, QUE FOI FUNDADA PARA TENTAR TRANSFORMAR O MODELO TMM EM UMA NORMA E, CONSEQUENTEMENTE, PROMOVER A SUA ACEITAÇÃO COMO UM PADRÃO DA INDÚSTRIA INTERNACIONAL DE AVALIAÇÃO E DE ORGANIZAÇÕES DE TESTE DE SOFTWARE. A FUNDAÇÃO TMMI TEM COMO OBJETIVO: A CRIAÇÃO E GESTÃO DE UMA ORGANIZAÇÃO INDEPENDENTE, IMPARCIAL COM REPOSITÓRIO CENTRAL DE DADOS E PRESTAÇÃO DE SERVIÇOS, MÉTODOS DE AVALIAÇÃO COM BASE NO MODELO PADRÃO, DEFINIÇÃO E MANUTENÇÃO DE AVALIADORES INDEPENDENTES E PRESTAÇÃO DE UM FÓRUM PÚBLICO DAS PARTES INTERESSADAS PARA FACILITAR A LIVRE TROCA DE INFORMAÇÃO, EDUCAÇÃO, IDÉIAS E USO DA NORMA PÚBLICA. 205

EM RELAÇÃO AO TPI, HÁ PESQUISAS NA ACADEMIA QUE OBJETIVAM MELHORAR A PRODUTIVIDADE DO TIME DE TESTES, UTILIZANDO AS PRÁTICAS DEFINIDAS PELA MELHORIA GRADUAL DO PROCESSO DE TESTES. 205

REFERÊNCIAS 206

11.2.2. ÁREAS DE CONHECIMENTO 209
REQUISITOS DE SOFTWARE 209
MANUTENÇÃO DE SOFTWARE 212
PROCESSO DE ENGENHARIA DE SOFTWARE 216
MÉTODOS E FERRAMENTAS DE ENGENHARIA 217
QUALIDADE DE SOFTWARE 218

REFERÊNCIAS 222

14.1. DEFINIÇÕES BÁSICAS 223

14.2. EVOLUÇÃO DO GERENCIAMENTO DE PROJETOS 225

14.3. RELEVÂNCIA DO GERENCIAMENTO DE PROJETOS 227

14.4. CONHECENDO O PMI 227

14.5. GERENCIAMENTO DE PROJETOS NA VISÃO DO PMI 228

14.6. TÓPICOS DE PESQUISA 228

14.7. SUGESTÕES DE LEITURA	228
14.8. EXERCÍCIOS	228
REFERÊNCIAS	228
GESTÃO DE PESSOAS	262
14.1 CONCEITOS E DESAFIOS	263
14.1.1 DESAFIOS ORGANIZACIONAIS PARA O GERENCIAMENTO DE PESSOAS.	263
14.1.1.1 AVANÇOS TECNOLÓGICOS	263
14.1.1.2 <i>DOWNSIZING</i>	264
14.1.1.3 CULTURA ORGANIZACIONAL	264
14.1.2 DESAFIOS INDIVIDUAIS	265
14.1.2.1 IDENTIFICAÇÃO COM A EMPRESA	265
14.1.2.2 CONDUTA ÉTICA	265
14.1.2.3 <i>EMPOWERMENT</i>	265
14.2 MOTIVAÇÃO: CONCEITOS E TEORIAS	266
14.2.2 TEORIAS DE MOTIVAÇÃO	267
14.2.3 PROCESSOS DE MOTIVAÇÃO	268
14.3 TRABALHO EM EQUIPE	270
14.3.2 TIPOS DE EQUIPE	270
14.4 GESTÃO DE PESSOAS POR COMPETÊNCIAS	271
14.4.2 REMUNERAÇÃO ESTRATÉGICA	273
14.4.3 AVALIAÇÃO DE DESEMPENHO	274
14.5 LIDERANÇA	275
14.5.1 O PAPEL DO LÍDER	275
14.5.2 <i>COACH</i>	276
14.6 GERENCIAMENTO DE CONFLITOS	277
14.6.1 VISÃO DOS CONFLITOS	277
14.6.2 NÍVEIS DE CONFLITO	277
14.6.3 CONFLITO E ESTRESSE	278
14.6.4 COMO GERIR CONFLITOS NO AMBIENTE DO PROJETO	278
ESTILO	278

DESCRIÇÃO	278
EFEITO	278
<i>RETIRAR-SE TEMPORARIAMENTE</i>	278
NÃO RESOLVE O PROBLEMA	278
<i>MUDAR O FOCO</i>	278
ENFATIZAR AS ÁREAS DE CONCORDANCIA EM DETRIMENTO DAS DIFERENÇAS	278
FORNECE APENAS UMA RESOLUÇÃO DE CURTO PRAZO	278
<i>EMPREGAR A FORÇA</i>	279
FORÇAR SEU PONTO DE VISTA INDEPENDENTEMENTE DOS OUTROS; OFERECE SOLUÇÕES DO TIPO GANHA-PERDE	279
SENTIMENTOS DE VINGANCA PODEM VOLTAR DE OUTRA MANEIRA	279
PROCURAR UMA TROCA QUE TRAGA ALGUM GRAU DE SATISFAÇÃO PARA TODAS AS PARTES	279
<i>RESOLVER O PROBLEMA</i>	279
TRATA CONFLITO COMO UM PROBLEMA A SEREM RESOLVIDOS, EXAMINANDO-SE TODAS AS POSSÍVEIS ALTERNATIVAS; REQUER DIALOGO ABERTO E ATITUDE PROATIVA DOS DOIS LADOS.	279
FORNECE UMA RESOLUÇÃO DE LONGO PRAZO	279
<i>NEGOCIAR</i>	279
INCORPORAR TODOS VÁRIOS PONTOS DE VISTA E VISÕES DAS DIFERENTES PERSPECTIVAS; CONDUZ A UM CONSENSO E COMPROMISSO.	279
FORNECE UMA SOLUÇÃO DE LONGO PRAZO	279
FONTE:????????????????????	279
14.7 GESTÃO DE PESSOAS E DESENVOLVIMENTO DA INTELIGÊNCIA EMOCIONAL	279
14.7.1 CONHECIMENTOS, HABILIDADES E ATITUDES (CHA)	280

14.8	CONSIDERAÇÕES FINAIS	281
14.9	SUGESTÕES DE LEITURA	282
14.10	TÓPICOS DE PESQUISA	284
14.11	EXERCÍCIOS	285
	REFERÊNCIAS	286
	VISÃO GERAL DA COMUNICAÇÃO	102
14.1.	PROCESSO DA COMUNICAÇÃO	103
14.1.1.	A COMUNICAÇÃO	103
	ATRAVÉS DAS HIERARQUIAS DE AUTORIDADE E ORIENTAÇÕES FORMAIS.	103
	INTEGRAÇÃO SOCIAL DENTRO DE GRUPOS SATISFAZENDO AS NECESSIDADES SOCIAIS.	104
	FORNECE SUBSÍDIOS PARA FACILITAR A TOMADA DE DECISÃO	104
14.1.4.	A COMUNICAÇÃO EM ORGANIZAÇÕES	105
	ATUALMENTE, O AMBIENTE ORGANIZACIONAL É CARACTERIZADO POR MUDANÇAS CONTÍNUAS. ASSIM, SURGINDO A NECESSIDADE DE MUDANÇA NOS MODELOS TRADICIONAIS DAS PRÁTICAS DA COMUNICAÇÃO ORGANIZACIONAL PARA MANTER A COMPETITIVIDADE EMPRESARIAL.	105
14.1.5.	COMUNICAÇÃO EM PROJETOS	106
14.1.6.	A COMUNICAÇÃO COMO DESAFIO PARA O GERENTE DE PROJETOS	108
14.2.	GERENCIAMENTO DE COMUNICAÇÃO EM PROJETOS	109
2.	FERRAMENTAS E TÉCNICAS PARA O PLANEJAMENTO DAS COMUNICAÇÕES:	113
3.	SAÍDAS DO PLANEJAMENTO DAS COMUNICAÇÕES:	114
	TEMPLATE DO PLANO DE COMUNICAÇÃO	115
1.	INTRODUÇÃO	115
2.	NECESSIDADES DE INFORMAÇÃO	115
3.	TIPOS DE INFORMAÇÃO	115

4. FORMATOS (TEMPLATES DE RELATÓRIOS)	115
--	------------

5. GLOSSÁRIO	115
---------------------	------------

14.2.2. DISTRIBUIÇÃO DAS INFORMAÇÕES	115
---	------------

14.2.3. RELATÓRIO DE DESEMPENHO	118
--	------------

14.2.4. GERENCIAR AS PARTES INTERESSADAS	123
---	------------

<u>EM PROJETOS DISTRIBUÍDOS, A COMUNICAÇÃO É A BASE PARA DEFINIR COMO SERÃO REPASSADAS AS INFORMAÇÕES PARA AS PARTES INTERESSADAS ENVOLVIDAS NO PROJETO. NÃO EXISTE UMA REGRA PARA GERENCIAR PROJETOS DISTRIBUÍDOS, MAS EXISTEM BOAS PRÁTICAS QUE SÃO PONTOS RELEVANTES E QUE AJUDAM OS PROJETOS A CHEGAREM A SEU OBJETIVO FUNDAMENTAL: SUA CONCLUSÃO NO PRAZO, DENTRO DO CUSTO E COM QUALIDADE. NA LITERATURA, PODEM SER ENCONTRADAS PESQUISAS E ARTIGOS COM ESTUDOS FOCADOS NESTE ASSUNTO.</u>	126
---	------------

REFERÊNCIAS	128
--------------------	------------

ALVES, A. A COMUNICAÇÃO NA GERÊNCIA DO PROJETO. REVISTA: TECHOJE: UMA REVISTA DE OPINIÃO. DISPONÍVEL EM: HTTP://WWW.IETEC.COM.BR/SITE/TECHOJE/CATEGORIA/ DETALHE_ARTIGO/101 . ACESSADO EM: SET. 2009.	128
---	-----

ARCANJO, C. (2008). CONTEXTO DA COMUNICAÇÃO NAS ORGANIZAÇÕES. DISPONÍVEL EM: HTTP://WWW.WEBARTIGOS.COM/ARTICLES/5381/1/CONTEXTO-DA-COMUNICACAO-NA-GESTAO-DAS-ORGANIZACOES/PAGINA1.HTML . ACESSADO EM: OUT. 2009.	128
--	-----

BARBOSA, L. O DESAFIO DA COMUNICAÇÃO EFICAZ NO GERENCIAMENTO DE PROJETOS. REVISTA: TECHOJE: UMA REVISTA DE OPINIÃO. DISPONÍVEL EM: HTTP://WWW.IETEC.COM.BR/SITE/TECHOJE/CATEGORIA/DETALHE_ARTIGO/61 . ACESSO EM: SET. 2009.	128
--	-----

CARVALHO, M.; MIRANDOLA, D. A COMUNICAÇÃO EM PROJETOS DE TI: UMA ANÁLISE COMPARATIVA DAS EQUIPES DE SISTEMAS E DE NEGÓCIOS, V.17 N.2, SÃO PAULO MAIO/AGO. 2007. DISPONÍVEL: HTTP://WWW.SCIELO.BR/SCIELO.PHP?SCRIPT=SCI_ARTTEXT&PID=S0103-65132007000200009&LNG=PT&NRM=ISO&TLNG=PT . ACESSADO EM: OUT. 2009.	128
---	-----

CASTELO, L. GERÊNCIA PARTICIPATIVA: A COMUNICAÇÃO E O GERENTE. DISPONÍVEL EM: HTTP://WWW.GERANEOCIO.COM.BR/HTML/GERAL/GP4.HTML . ACESSADO EM: SET. 2009.	128
---	-----

JACOB, M. IMPORTÂNCIA DA COMUNICAÇÃO NA GERÊNCIA DE PROJETOS. REVISTA: TECHOJE: UMA REVISTA DE OPINIÃO. DISPONÍVEL EM: HTTP://WWW.IETEC.COM.BR/SITE/TECHOJE/CATEGORIA/DETALHE_ARTIGO/100 . ACESSADO EM: SET. 2009.	129
--	-----

PIMENTA, J. A COMUNICAÇÃO NAS EMPRESAS E EM PROJETOS. REVISTA: TECHOJE: UMA REVISTA DE OPINIÃO. DISPONÍVEL EM: HTTP://WWW.IETEC.COM.BR/SITE/TECHOJE/CATEGORIA/ DETALHE_ARTIGO/691 . ACESSADO EM: OUT. 2009.	130
---	-----

PMI (PROJECT MANAGEMENT INSTITUTE) A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE – GUIA PMBOK® 4. ED. UPPER DARBY, 2008.	130
--	-----

RIVAS, M. PLANEJAMENTO & COMUNICAÇÃO PARA ESTABELECEER UM DIFERENCIAL COMPETITIVO. REVISTA: TECHOJE: UMA REVISTA DE OPINIÃO. DISPONÍVEL EM: HTTP://WWW.IETEC.COM.BR/SITE/TECHOJE/CATEGORIA/DETALHE_ARTIGO/379 . ACESSADO EM: SET. 2009.	130
---	-----

<u>SCHNEIDER, G. (2008) O GERENTE DE PROJETOS TAMBÉM CUIDA DA COMUNICAÇÃO. WEBINSIDER. DISPONÍVEL EM: HTTP://WEBINSIDER.UOL.COM.BR/INDEX.PHP/2008/11/05/O-GERENTE-DE-PROJETOS-TAMBEM-CUIDA-DA-COMUNICACAO/. ACESSADO EM: SET. 2009.</u>	131
---	------------

15.1. IMPORTÂNCIA DA MEDIÇÃO	103
-------------------------------------	------------

15.2. O QUE SÃO MÉTRICAS **104**

REFERÊNCIAS **106**

17 GESTÃO DE PROGRAMAS **109**

▪ PROGRAMAS **109**

▪ GERENCIAMENTO DE PROGRAMAS	111
RELAÇÃO ENTRE GERENCIAMENTO DO PROGRAMA E GERENCIAMENTO DO PROJETO	111
TEMAS DO GERENCIAMENTO DE PROGRAMA	112
• GERENCIAMENTO DE BENEFÍCIOS	112
• GERENCIAMENTO DE <i>STAKEHOLDERS</i>	113
• GOVERNANÇA	114
CICLO DE VIDA DO PROGRAMA	115
• FASE 1: SET UP PRÉ-PROGRAMA	116
• FASE 2: SET UP PROGRAMA	117
• FASE 3: ESTABELECEER ESTRUTURA DE GESTÃO DO PROGRAMA	118
• FASE 4: BENEFÍCIOS INCREMENTAIS	118
• FASE 5: ENCERRAMENTO	119
▪ PROCESSOS DO GERENCIAMENTO DE PROGRAMA	119
GRUPO PROCESSOS DE INICIAÇÃO	120
GRUPO PROCESSOS DE PLANEJAMENTO	121
GRUPO PROCESSOS DE EXECUÇÃO	123
GRUPO PROCESSOS DE MONITORAMENTO E CONTROLE	124
GRUPO PROCESSOS DE ENCERRAMENTO	125

▪ TÓPICOS DE PESQUISA **126**

▪ SUGESTÕES DE LEITURA **126**

▪ EXERCÍCIOS **126**

▪ REFERÊNCIAS **127**

GESTÃO DE PORTFÓLIO DE PROJETOS **103**

INTRODUÇÃO	103
DEFINIÇÃO DE PORTFÓLIO	104
ESTRATÉGIA CORPORATIVA E GESTÃO DE PORTFÓLIO	104
GESTÃO DE PORTFÓLIO VERSUS GESTÃO DE MÚLTIPLOS PROJETOS	106
RELAÇÃO ENTRE A GESTÃO DE PORTFÓLIO E A GESTÃO DE PROJETOS/PROGRAMAS	106
MÉTRICAS EM GESTÃO DE PORTFÓLIO [PMI 2006]	106
GERENTE DE PORTFÓLIO [PMI 2006]	107
MODELOS E PADRÕES DE GESTÃO DE PORTFÓLIO	107
ADRÃO DE GESTÃO DE PORTFÓLIO [PMI 2006]	107
PROCESSO STAGE-GATE [COOPER ET AL 2001]	111
PROCESSO INTEGRADO DE SELEÇÃO E PRIORIZAÇÃO DE PROJETOS [ARCHER AND GHASEMZADEH 1999]	113

ESTUDO DE CASO: GESTÃO DE PORTFÓLIO DE PROJETOS NO SERPRO	114
SUGESTÕES DE LEITURA	115
TÓPICOS DE PESQUISA (TRABALHOS FUTUROS E CORRENTES)	116
O IMPACTO DA GESTÃO DE PORTFÓLIO DE PROJETOS EM PROJETOS DE TECNOLOGIA DA INFORMAÇÃO [REYCK ET AL. 2005]	116
PORTFOLIUS: UM MODELO DE GESTÃO DE PORTFÓLIO DE PROJETOS DE SOFTWARE [CORREIA 2005]	116
SELEÇÃO DE PROJETOS EM UM PORTFÓLIO PARA APOIO A TOMADA DE DECISÃO [GHASEMZADEH AND ARCHER 2000]	116
UM PROCESSO INTEGRADO PARA SELEÇÃO DE PROJETOS EM UM PORTFÓLIO [ARCHER AND GHASEMZADEH 1999]	116
EXERCÍCIOS	116
REFERÊNCIAS BIBLIOGRÁFICAS	116
1.1 INTRODUÇÃO	102
1.2 PAPÉIS E FUNÇÕES	103
1.3 OBJETIVOS DE UM PMO	104
1.4 TIPOS DE PMOS	105
1.4.1 KERZNER	105
1.4.2 DINSMORE E VARGAS	105
1.4.3 CRAWFORD	110
1.5 BOAS PRÁTICAS NA IMPLANTAÇÃO DE PMOS	111
1.6 ESTUDO DE CASO: A IMPLANTAÇÃO DE ESCRITÓRIO DE PROJETOS NO SERPRO	113
1.6.1 O SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS – SERPRO	113
1.6.2 MOTIVAÇÃO	114
1.6.3 IMPLANTAÇÃO	115
1.6.3.1 ESTRATÉGIA	115
1.6.3.2 FASES	116
1.6.4 BENEFÍCIOS ALCANÇADOS	117
1.6.5 MELHORIA CONTÍNUA	117
1.7 TÓPICOS DE PESQUISA	118

1.8 SUGESTÕES DE LEITURA **119**

DURANTE A CONSTRUÇÃO DESTES CAPÍTULOS FORAM IDENTIFICADAS ALGUMAS SUGESTÕES DE LEITURA QUE PODEM AJUDAR O LEITOR A MELHOR COMPREENDER O CONTEXTO DE ESCRITÓRIO DE PROJETOS. **119**

ESTAS LEITURAS SÃO LISTADAS A SEGUIR: **119**

= PARA UM MAIOR DETALHAMENTO SOBRE O GESTÃO DE PROJETOS LER: **119**

= A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE [PMI 2004]; **119**

= ADVANCED PROJECT MANAGEMENT: BEST PRACTICES ON IMPLEMENTATION [KERZNER 2004] **119**

= E PROJECT MANAGEMENT: A SYSTEMS APPROACH TO PLANNING, SCHEDULING, AND CONTROLLING [KERZNER 2006]. **119**

= PARA CONHECER UM POUCO MAIS SOBRE A IMPORTÂNCIA E TER UMA VISÃO GERAL SOBRE ESCRITÓRIO DE PROJETOS LER: **119**

= OS CAÇADORES DE PROJETOS. EXTRAÍDO DA REVISTA EXAME EXAME DE 17/08/2004. CRISTIANE MANO. **119**

HTTP://WWW.PMTECH.COM.BR/ARTIGOS/PMO_PMTECH_V1.PDF **119**

= PARA UM MELHOR ENTENDIMENTO AS ESTRATÉGIAS PARA A IMPLEMENTAÇÃO DE UM ESCRITÓRIO DE PROJETOS LER: **119**

= IMPLEMENTANDO UM ESCRITÓRIO DE PROJETOS. RICARDO MANSUR. 1ª EDIÇÃO – 2007. HTTP://WWW.BRASPORT.COM.BR/INDEX.PHP?ESCOLHA=8&LIVRO=L00233 **119**

= IMPLANTANDO O ESCRITÓRIO DE GERENCIAMENTO DE PROJETOS (PMO). **119**

HTTP://WWW.PMTECH.COM.BR/ARTIGOS/PMO_PMTECH_V1.PDF **119**

= PARA CONHECER UM POUCO MAIS O RELACIONAMENTO ENTRE ESCRITÓRIO DE PROJETOS E RETORNO DE INVESTIMENTO NAS ORGANIZAÇÕES LER: **119**

= ESCRITÓRIO AVANÇADO DE PROJETOS NA PRÁTICA. RICARDO MANSUR. 1ª EDIÇÃO – 2009. HTTP://WWW.BRASPORT.COM.BR/INDEX.PHP?ESCOLHA=8&LIVRO=L00307. **119**

= PARA CONHECER SOBRE ABORDAGENS PARA IDENTIFICAÇÃO DE FATORES CRÍTICOS NA IMPLANTAÇÃO DE ESCRITÓRIO DE PROJETOS LER: **119**

– PARA CONHECER O RELACIONAMENTO ENTRE ESCRITÓRIO DE PROJETOS E GESTÃO DO CONHECIMENTO LER: **120**

1.9 EXERCÍCIOS **121**

– DESCREVA OS PRINCIPAIS GRUPOS DE FUNÇÕES APRESENTADOS NA PESQUISA DE HOBBS E AUBRY? **121**

– CITE OS PRINCIPAIS OBJETIVOS DOS PMOS. **121**

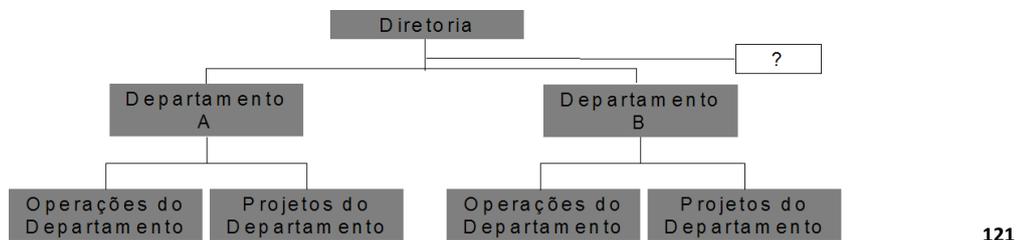
– DESCREVA E DIFERENCIE OS 3 (TRÊS) TIPOS DE PMO’S DEFINIDOS SEGUNDO KERZNER. **121**

– DESCREVA E DIFERENCIE OS 5 (CINCO) TIPOS DE PMO’S DEFINIDOS SEGUNDO DINSMORE. **121**

– DESCREVA E DIFERENCIE OS 3 (TRÊS) NÍVEIS DE PMO’S CARACTERIZADOS POR CRAWFORD. **121**

– ANALISANDO A FIGURA ABAIXO, A QUAL REPRESENTA UM ORGANOGRAMA ORGANIZACIONAL DE UMA EMPRESA FICTÍCIA, QUE TIPO ESCRITÓRIO DE PROJETOS MELHOR SE ENQUADRARIA SEGUNDO A CAIXA DESTACADA EM CINZA? **121**

– ANALISANDO A FIGURA ABAIXO, A QUAL REPRESENTA UM OUTRO ORGANOGRAMA ORGANIZACIONAL DE UMA SEGUNDA EMPRESA FICTÍCIA, QUE TIPO ESCRITÓRIO DE PROJETOS MELHOR SE ENQUADRARIA SEGUNDO A CAIXA DESTACADA EM CINZA? **121**



REFERÊNCIAS **122**

20.1. INTRODUÇÃO A MATURIDADE EM GESTÃO DE PROJETOS **102**

20.2. MODELOS DE MATURIDADE EM GESTÃO DE PROJETOS **103**

20.2.1. ORGANIZATIONAL PROJECT MANAGEMENT MATURITY MODEL - PMI **103**

20.2.2. PROJECT MANAGEMENT MATURITY MODEL – PM SOLUTIONS **104**

20.2.3. MODELO DE MATURIDADE EM GERENCIAMENTO DE PROJETOS – DARCI PRADO **106**

20.2.4. PORTFOLIO, PROGRAMME AND PROJECT MANAGEMENT MATURITY MODEL – OGC **106**

20.2.5. KERZNER PROJECT MANAGEMENT MATURITY MODEL – HAROLD KERZNER **107**

20.3. OPM3 **108**

20.3.1. ESTRUTURA DO MODELO	108
20.3.2. AVALIAÇÃO DA MATURIDADE	109
20.3.3. IMPLANTAÇÃO DO MODELO	110
<u>20.4 MMGP</u>	<u>112</u>
20.4.1. ESTRUTURA DO MODELO	112
20.4.2. AVALIAÇÃO DA MATURIDADE	113
20.4.3. IMPLANTAÇÃO DO MODELO	114
<u>20.5. KPMMM</u>	<u>114</u>
20.5.1. ESTRUTURA DO MODELO	114
20.5.2. AVALIAÇÃO DA MATURIDADE	116
20.5.3. IMPLANTAÇÃO DO MODELO	120
<u>20.6. UM ESTUDO DE CASO</u>	<u>121</u>
20.6.1. METODOLOGIA	122
20.6.2. RESULTADOS COLETADOS	122
20.6.3. PERFIL DOS PARTICIPANTES	123
20.6.4. SEGMENTAÇÃO POR NÍVEL DE MATURIDADE	125
20.6.5. SEGMENTAÇÃO POR PERCENTUAL DE ADERÊNCIA AOS NÍVEIS DE MATURIDADE	126
20.6.6. CONCLUSÃO	129
<u>20.7. ANÁLISE COMPARATIVA</u>	<u>129</u>
<u>20.8. SUGESTÕES DE LEITURA</u>	<u>101</u>
<u>20.9. TÓPICOS DE PESQUISA</u>	<u>101</u>
<u>20.10. EXERCÍCIOS</u>	<u>101</u>
<u>REFERÊNCIAS</u>	<u>102</u>
<u>GOVERNANÇA EM TIC</u>	<u>106</u>
<u>GESTÃO EM TIC</u>	<u>106</u>
RELEVÂNCIA E EVOLUÇÃO DO PAPEL DA TIC NAS ORGANIZAÇÕES	108
DA GESTÃO À GOVERNANÇA EM TIC	110
<u>MODELOS DE GESTÃO EM TIC</u>	<u>112</u>
COBIT	112
ITIL	113

BSC	113
IT FLEX	113
COSO	114
ISO/IEC 20000	115
VAL IT	115
CMMI SOB A PERSPECTIVA DE GOVERNANÇA DE TI	115

ITIL **116**

DEFINIÇÃO	116
HISTÓRICO	116
REGULAMENTAÇÃO DO ITIL	117
1. DIREITOS AUTORAIS	117
2. CERTIFICAÇÕES / TREINAMENTOS	117
3. PUBLICAÇÃO DE CONTEÚDOS OFICIAIS	118
4. FÓRUM DE FOMENTO (ITSMF)	118
ESTRUTURA DO ITIL	118
• <i>SERVICE STRATEGY</i> (ESTRATÉGIA DE SERVIÇOS)	118
• <i>SERVICE DESIGN</i> (PLANEJAMENTO DE SERVIÇOS)	118
• <i>SERVICE TRANSITION</i> (TRANSIÇÃO DE SERVIÇOS)	118
• <i>SERVICE OPERATION</i> (OPERAÇÃO DE SERVIÇOS)	119
• <i>CONTINUAL SERVICE IMPROVEMENT</i> (APRIMORAMENTO CONTÍNUO DE SERVIÇOS)	119
FRONTEIRAS COM OUTROS MODELOS E LIMITAÇÕES	121
PONTO DE PARTIDA	121
COMENTÁRIOS SOBRE PRÁTICAS DE SUCESSO	122
PÚBLICO ALVO	123
UTILIZAÇÃO DO ITIL	124

COBIT **125**

DEFINIÇÃO	125
HISTÓRICO	125
REGULAMENTAÇÃO DO COBIT	126
5. CERTIFICAÇÕES / TREINAMENTOS	126
6. DIREITOS AUTORAIS	126
7. PUBLICAÇÃO DE CONTEÚDOS OFICIAIS	127
8. FÓRUM DE FOMENTO (ISACA)	127
ESTRUTURA DO COBIT	127
9. PRIMEIRA DIMENSÃO DO CUBO – PROCESSOS DE TI	128
10. SEGUNDA DIMENSÃO DO CUBO – CRITÉRIOS DE INFORMAÇÃO	130
11. TERCEIRA DIMENSÃO DO CUBO – RECURSOS DE TI	131
NÃO É COBIT	131
FRONTEIRAS COM OUTROS MODELOS	132
PONTO DE PARTIDA	133
COMENTÁRIOS SOBRE PRÁTICAS DE SUCESSO	134
PÚBLICO ALVO	134
UTILIZAÇÃO DO COBIT	134

INICIATIVAS DE INTEGRAÇÃO DOS PRINCIPAIS MODELOS **135**

IMPLANTAÇÃO DE MODELOS DE GESTÃO	136
TÓPICOS DE PESQUISA	138
SUGESTÕES DE LEITURA	139
EXERCÍCIOS	140
REFERÊNCIAS	142

Capítulo

1

[VERSÃO DRAFT 11 de novembro de 2009]

Processos tradicionais de desenvolvimento de software

Wislayne Aires Moreira

1.1 Introdução

Os processos tradicionais de desenvolvimento de software são processos onde se podem presumir todos os requisitos do sistema, que traz a vantagem de tornar os projetos completamente planejados, facilitando a gerência do mesmo, mantendo sempre uma linha, caracterizando o processo como bastante rigoroso. [Galdino, C, 2008]. Os processos orientados a documentos são considerados rigorosos, devido à especificação de requisitos o que torna essa etapa importante para o cliente, onde as suas necessidades são documentadas e definidas. Os processos tradicionais são caracterizados por possuir abordagem voltada ao uso de documentação detalhada das atividades, fases seqüenciais de processo e um conjunto de artefatos em cada fase, bons para atividades estáveis, mas caro para ambientes dinâmicos, exige pessoal experiente e o ambiente funciona na organização de responsabilidades. Podemos citar alguns processos tradicionais existentes: RUP, OpenUp, EUp, PSP, TSP, MSF, Catalisys, ICONIX e etc.

Comment [MA1]: Iniciais em maiúsculo

Comment [MA2]: O professor tinha recomendado fazer tipo uma apresentação de poucas linhas do capítulo antes de uma introdução mesmo ao tema, seria isso?

Comment [MA3]: Verificar o padrão [FULADO 2009]

Comment [MA4]: As primeiras citações de siglas devem vir com o respectivo significado, tipo Rational Unified Process (RUP)

Ao longo deste capítulo serão apresentados, os processos tradicionais de software mais utilizados como: RUP (Rational Unified Process), o processo mais difundido do Brasil e o mais utilizado por empresas que possuem desenvolvimento de software; OpenUp (Open Unified Process) e MSF (Microsoft Solutions Framework), as suas origens, o significado de cada processo e características, a arquitetura, as suas etapas, disciplinas e ciclo de vida dos processos.

Comment [MA5]: Retirar “;”
Comment [MA6]: Ao longo do texto ter cuidado com o espaço entre palavras e pontuação

1.2 RUP

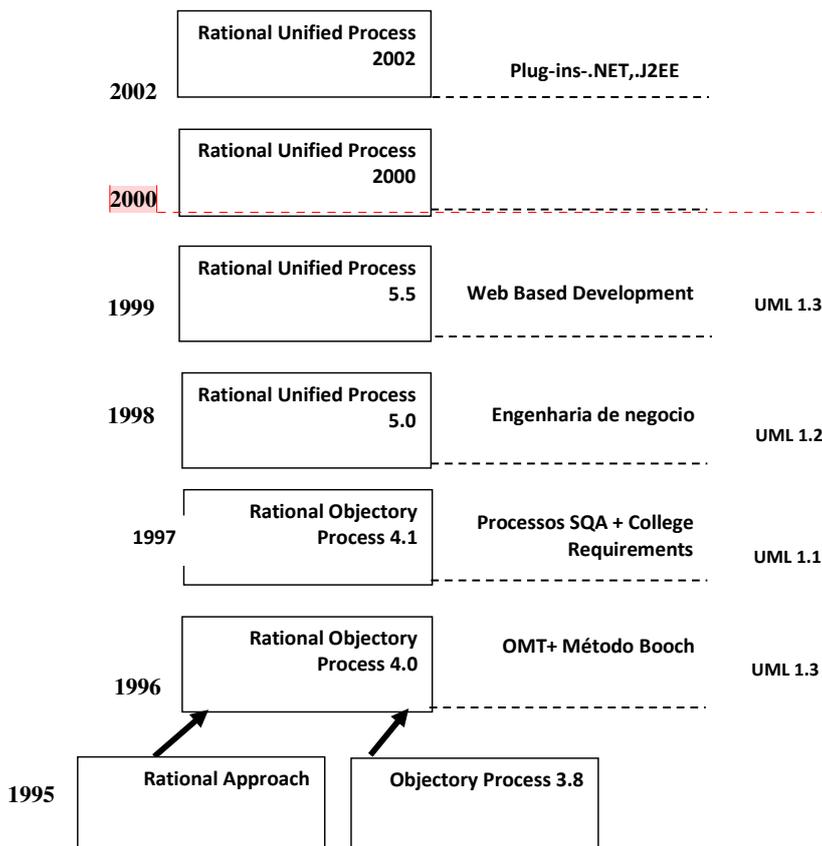
1.2.1 Origem do RUP

É um processo de engenharia de software que foi patenteado pela Rational Software Corporation, adquirida pela IBM tornando-se uma marca na área de software, na qual fornece técnicas a serem seguidas pelos integrantes da equipe que desenvolve software com a finalidade de aumentar a produtividade da equipe.

Comment [MA7]: Entre seções e subseções eu sempre costumo fazer algum comentário, nesse caso poderia ou fazer um comentário entre RUP e a origem ou tirar a subseção origem do RUP

O Rational Unified Process (RUP) surgiu em 1998, com abordagens seguidas na Ericson, onde trabalhou Ivan Jacobson (1967), que mais tarde, se uniu a Grady Booch e James Rumbaugh, denominado “os três amigos”, começaram as iniciativas para a unificação de suas metodologias, desenvolvidas desde 1981 na Rational. A evolução do RUP pode ser mostrada na figura abaixo:

Comment [MA8]: Ajustar parágrafo e margens, só o parágrafo inicial da seção não possui recuo



Comment [MA9]: A imagem está muito no canto direito, já que todas as informações estão em caixas de texto, agrupa tudo e transforma em uma única, fica melhor pra manipular

Figura 1.1. História do RUP

1.2.2 O RUP e suas características

O Rational Unified Process® (também chamado de processo RUP®) é um método de desenvolvimento iterativo e incremental. É um processo iterativo por que um grande projeto é dividido em projetos menores, que após término tem-se um produto acabado, que serve para ser utilizado em outra iteração e é incremental por que em cada iteração e é incremental por que cada iteração é um incremento, que é o crescimento do produto. O RUP é definido em três elementos centrais que são [Dantas, F, 2003]:

- Uma abordagem de desenvolvimento de software iterativo, centrado em arquitetura e caso de uso baseado em risco;
- Um processo de engenharia de software bem definido e estruturado;
- Um produto do processo que fornece uma estrutura customizável do processo.

O RUP é um processo que se caracteriza pelos seguintes elementos: a UML é uma parte integrante do RUP; o RUP e UML foram desenvolvidos juntos; centrado em uma arquitetura, onde promove a definição inicial de uma arquitetura robusta, que facilita a paralelização do desenvolvimento, reutilização e a manutenção e guiados por caso de uso.

1.2.3 Visão Geral do RUP

A arquitetura é dividida em duas estruturas, as quais refletem as duas visões em que um sistema pode ser descrito: componentes dinâmicos e componentes estáticos.

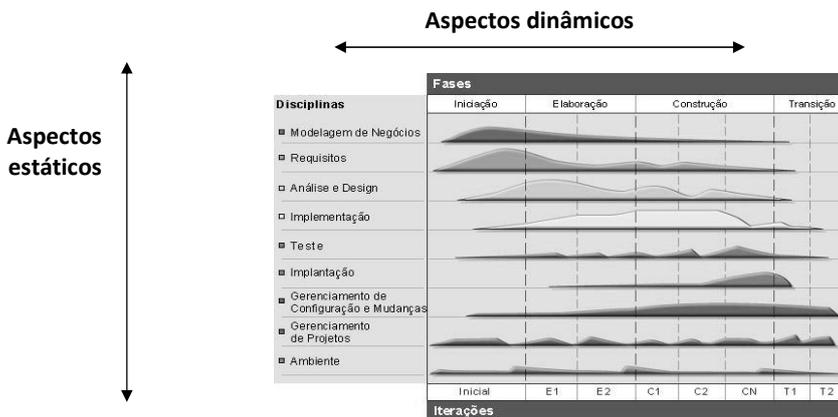


Figura 1.2 Arquitetura RUP

A dimensão horizontal representa a estrutura dinâmica ou tempo dos processos. Os processos em termos de ciclo, fases (concepção, elaboração, construção e transição), iterações e milestones. [Dantas, F, 2003, Rational Software, 2001]

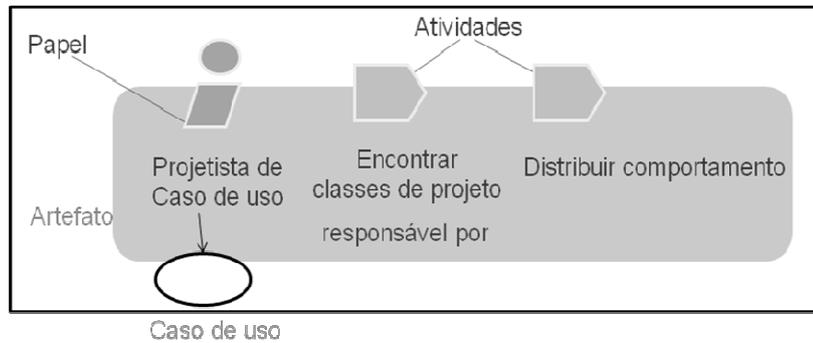
A dimensão vertical representa a estrutura estática do processo. Descreve como elementos do processo (atividades, disciplinas, artefatos e papéis) são agrupados em disciplinas de processo ou workflows. [Dantas, F, 2003]. Na estrutura estática um processo descreve “quem” está fazendo “o quê”, “como” e “quando”, conforme demonstrado abaixo:

Comment [MA10]: Necessita de um distanciamento entre a seção e o início dela

Comment [MA11]: Acho que isso @ não será preciso, será?

Comment [MA12]: Tirar recuo

Comment [MA13]: Colocar recuo nos parágrafos posteriores ao primeiro da seção



Comment [MA14]: Fonte ou adaptação de: ...

Figura 1.3 Descrição do processo

- Papéis - representam quem será responsável por determinada função (tarefa);
- Atividades - representa que unidade de trabalho que um individuo em um papel pode ser questionado a executar;
- Artefatos - representa os elementos tangíveis de um projeto. Pode assumir várias formas o como: modelo, documento, código fonte ou executável;
- Fluxos de trabalho - representa o quando. Uma forma de descrever seqüência significativa de atividades que produzem algum artefato e que mostram iterações entre os papéis.

1.2.4 Fases do RUP

O ciclo de desenvolvimento do RUP é dividido em 4 fases que são: Concepção, Elaboração, Construção e Transição. O ciclo de vida termina com a entrega do produto final. Um erro que freqüentemente acontece é que, apesar das fases serem seqüenciais, as pessoas as tratam com processo em cascata, onde a primeira fase deve ser encerrada antes de começar a próxima. É interessante lembrar que o RUP é iterativo e baseado em riscos. [Dantas, F, 2008]

A finalidade das fases do RUP não é o de dividir as atividades por tipo (análise, implementação e etc.). Isso já é algo que é concebido pelo conceito de disciplina. O real objetivo das fases é o de fazer o suficiente para que uma determinada atividade atinja os seus objetivos da fase e ao mesmo tempo conclua os milestones. O que faz com que chegue aos objetivos de cada etapa são os riscos envolvidos nas mesma, por exemplo:

- Na fase de Concepção temos que analisar se o projeto compensa financeiramente;
- Elaboração - O foco é nos riscos técnicos, examinando a arquitetura revendo o escopo para que os requisitos fiquem claros;
- Construção - Lida com os riscos lógicos do projeto. Está relacionado à pessoas e infra-estrutura;
- Transição - O risco está associado com a entrega do produto ao usuário final.

1.2.4.1 Concepção

Para quem é iniciante no RUP pode querer desistir, devido a sua complexidade. é por isso que tem que ter um bom entendimento dos objetivos das fases do RUP e analisar o que cada objetivo representa no contexto.

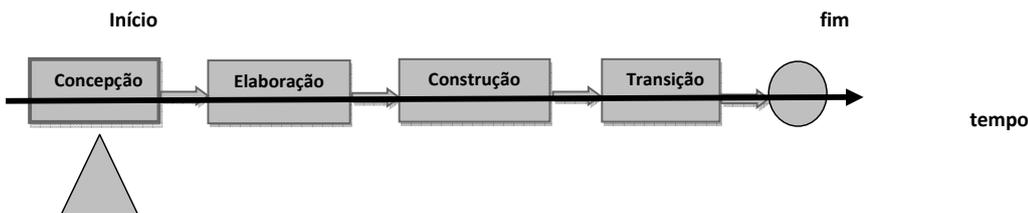


Figura 1.4 Fase de concepção

Na fase de Concepção o foco é em entender o escopo e os objetivos do projeto, daí obter informações suficientes para se decidir se vai seguir com o projeto ou abortá-lo. Os principais objetivos são:

1. Compreender o que vai construir, determinar o escopo do sistema, entendimento do escopo do sistema por todos os *stakeholders*. Para um entendimento de todas as pessoas no projeto é necessário algumas atividades como:

- Produção de um documento de visão: Esse documento de revisão pode estar pronto na fase de Concepção, mas pode ser refinado ao longo do projeto. Deve apresentar claramente aos *stakeholders* os benefícios do sistema, problemas que serão resolvidos pela aplicação, os principais usuários do sistema e os principais requisitos não funcionais. Nessa fase tem de detalhar os atores e casos de uso centrais, assim como gerar protótipos de interface com o usuário que permitirá a validação dos fluxos de eventos com os *stakeholders* e identificar as funcionalidades centrais do sistema.

Neste momento poderá ser necessário a implementação de uma parte da arquitetura para que os riscos sejam aceitáveis.

Entender o que vai desenvolver é importante, mas determinar os custos e prazos é crucial para desenvolver um projeto

Compartilhar uma visão de como o produto será desenvolvido, ou seja, qual o processo será utilizado.

1.2.4.2 Elaboração

Essa fase as diferenças com o modelo cascata fica evidente em relação ao modelo iterativo, onde apresenta as vantagens do modelo iterativo.

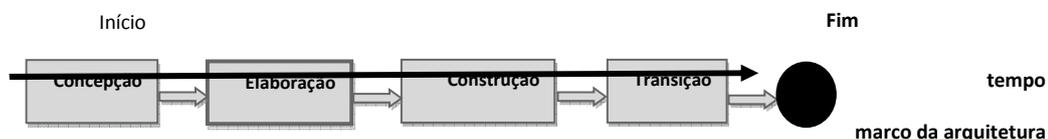


Figura 1.5 Fase de elaboração

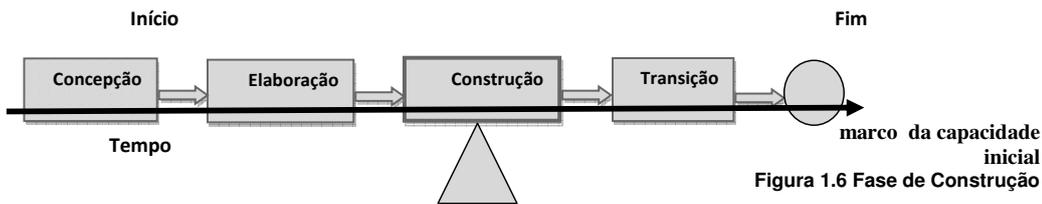
O objetivo desta fase é definir e consolidar uma arquitetura de sistema provendo uma base estável para as atividades de design e implementação. Os riscos associados a esta fase estão relacionados a requisitos, arquitetura, custo e cronograma e processo e ferramentas. Estes riscos serão analisados abaixo:

- Aos requisitos- tem que analisar se a aplicação está correta;
- Na arquitetura- tem que analisar se a solução correta está sendo construída;
- Custos e cronogramas- se realmente você está no caminho certo;
- Processo e ferramentas- Se possuem o processo e ferramentas corretos para fazer o trabalho. Quando não há o conhecimento do domínio do sistema, esse objetivo poderá requer mais iterações para que possa ser atingido e assim os riscos mitigados.

A fase de elaboração é onde a arquitetura do software na qual os requisitos que mais chocam a arquitetura são capturados em forma de caso de uso. Há identificação dos riscos que o projeto proporciona e ao final dessa fase deve-se possível estimar custos, elaborar cronogramas e plano de construção do sistema. Está é a fase mais crítica, porque é nessa fase que a engenharia é considerada completa e os custos para modificação do sistema aumentam a medida que o projeto avança. É nessa etapa que o projeto passa de custos e riscos baixos para custos e riscos altos.

1.2.4.3 Construção

Essa fase está relacionada aos riscos “lógicos”, e a maior parte do trabalho é realizado. É a etapa de construção do produto.



Os objetivos principais dessa etapa são:

- Diminuir os custos de desenvolvimento, otimizar os recursos e evitar retrabalho desnecessário;
- Concluir a análise, o projeto, o desenvolvimento e o teste de todas as funcionalidades necessárias;
- Desenvolver o modelo iterativo e incremental do produto completo que esteja pronto para a transição para a comunidade de usuários. Implica também em descrever os casos de uso restantes e outros requisitos, incrementar o projeto, concluir a implementação e testar o software;
- Decidir se o software, os locais e os usuários estão prontos para que o aplicativo seja implementado;
- Atingir certo paralelismo entre o trabalho das equipes de desenvolvimento.

As atividades básicas da fase de construção são:

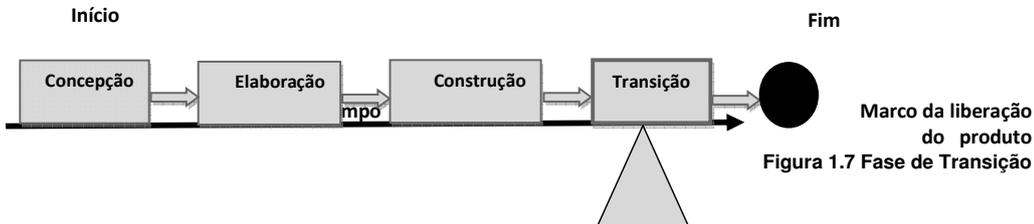
- Gerenciamento de recursos, otimização de controle e processo;
- Desenvolvimento completo do componente e teste dos critérios de avaliação definidos e a avaliação dos releases do produto de acordo com os critérios de aceitação para a visão.

Os critérios de avaliação para a etapa de construção envolvem as seguintes perguntas:

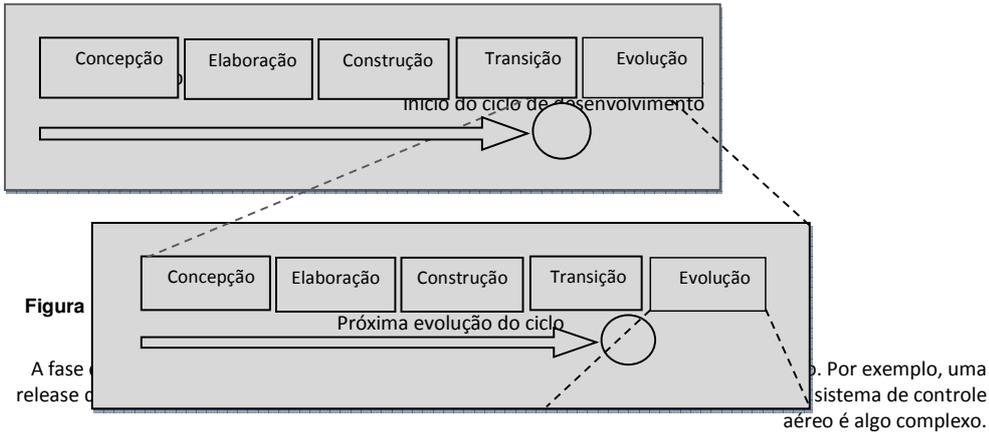
- Este release do produto é estável e desenvolvido o suficiente para ser implantado na comunidade de usuários?
- Todos os envolvidos estão prontos para a transição com a comunidade de usuários?
- As despesas reais com recursos ainda são aceitáveis se comparadas com as planejadas?

1.2.4.4 Transição

Nessa fase, o sistema está pronto. Começa a implantação do sistema para o usuário final. O foco nessa fase é garantir que o software esteja disponível para os usuários finais.



No final do ciclo de vida da fase de transição, os objetivos devem ter sido atendidos e o projeto deve estar concluído. Em alguns casos o ciclo de vida atual pode combinar com o início de outro ciclo de vida, levando a uma próxima geração do mesmo produto. O ciclo de desenvolvimento é quando o processo passa pelas 4 fases do RUP e os ciclos sucessivos é chamado de evolução de ciclos.



As atividades que serão realizadas nas iterações dependem muito da meta que se deseja. Por exemplo, para corrigir erros do sistema, normalmente utilizo as atividades de implementação e teste. As atividades básicas dessa fase são: realizar planos de implantação, testar o produto final, criar release do produto, conseguir *feedback* do usuário, concordar o produto com base no *feedback* e disponibilizar o produto para o usuário final.

1.2.5 Princípios básicos do RUP

Os seis princípios mais importantes do RUP:

a) Desenvolver software iterativamente

Permite entregar um sistema completo e aumenta a funcionalidade de cada subsistema a cada *release*.

b) Gerenciar requisitos

O RUP descreve como documentar a funcionalidade, restrições de sistema, restrições de projeto e requisitos de negócio.

Os casos de uso cenários são exemplos de artefatos que são dependentes do processo que são muito eficazes na captura de requisitos funcionais.

c) Usar arquiteturas baseadas em componentes

A arquitetura baseada em componentes cria um sistema que pode ser facilmente extensível, requerendo a reutilização de software e um entendimento intuitivo.

O RUP oferece uma forma metódica para construir esse sistema, focando-se em produzir uma arquitetura executável nas fases iniciais do projeto, ou seja, antes de comprometer recursos em larga escala.

d) Modelar visualmente o software

Comment [MA15]: Formato: justificado

O uso de modelos visuais pode permitir aos clientes que tenham um melhor entendimento de um dado problema, e dessa forma possam se envolver mais no projeto como um todo. A linguagem UML é amplamente utilizada no RUP para representar projetos.

e) Verificação contínua da qualidade

Não ter a preocupação com qualidade é uma falha muito comum em projetos computacionais. As pessoas envolvidas em um projeto estão preocupada com a qualidade do software após o término dos projetos, ou acham que a qualidade é responsabilidade de grupo específico, diferente da equipe de desenvolvimento. RUP visa auxiliar no controle do planejamento da qualidade, analisando a construção de todo o processo e envolvendo todos os membros da equipe de desenvolvimento.

f) Controle de mudanças

Em todos os projetos de software a existência de mudanças é inevitável. O RUP define métodos para controlar e monitorar mudanças. Como uma pequena mudança pode afetar aplicações de formas inteiramente imprevisíveis, o controle de mudanças é essencial para o sucesso de um projeto.

O RUP também define áreas de trabalho seguras, garantindo a um programador que as mudanças efetuadas em outro sistema não afetarão o seu sistema.

1.2.6 Disciplinas

Uma disciplina no RUP expressa todas as atividades que devem ser realizadas para produzir um conjunto de artefatos.

O RUP organiza suas disciplinas da seguinte forma:

- Fluxos de processo correspondem às atividades de desenvolvimento: modelagem de negócios, requisitos, análise e projeto, implementação, testes e implantação.
- Fluxos de suporte correspondem às atividades de gerenciamento e infra-estrutura: gerenciamento de configuração e mudanças, gerenciamento de projeto e ambiente.

As disciplinas Fluxos de processo são divididas em 9, que são elas:

- **Modelagem do negócio:** Os objetivos desta disciplina são de compreender a estrutura e a dinâmica da organização-alvo; entender os problemas da organização; fazer com que os clientes, desenvolvedores tenham entendimento comum em relação à organização e obter os requisitos necessários do sistema para sustentar a organização.

A disciplina de Modelagem e Negócios é baseada em modelos de casos de uso de negócios e um modelo de objeto de negócios. O modelo de caso de uso de negócio descreve uma sequência de atividades necessárias para fornecer um resultado para o ator de negócio e o modelo de objetos de negócio descreve os casos de uso de negócios.

- **Requisitos:** Na disciplina de requisitos o foco é o de manter concordância entre os clientes e as partes interessadas no que diz respeito aquilo que o sistema deve fazer; fornecer aos desenvolvedores uma melhor compreensão dos requisitos do sistema; ter uma base para estimar custos e prazos de desenvolvimento do sistema e definir uma interface para o usuário ter um entendimento de como o sistema funcionará de acordo com as suas necessidades.

Os modelos de caso de uso de negócios e objetos de negócio utilizado na disciplina de Modelagem e Negócio servirão de informações importantes para esse fluxo. Além desses documentos, serão criados um documento de visão, modelo de caso de uso, casos de uso e especificação suplementar, isso para descrever o que o sistema fará, onde todos os envolvidos são fontes de informações. Um documento de visão descreve qual o entendimento que as partes envolvidas do sistema têm dele. Modelo de casos de uso são é o modelo das funções pretendidas do sistema e seu ambiente. Casos de uso é uma sequência de ações que resultem em valor para um ator específico. Especificação

suplementar captura os requisitos que não são capturados pelos casos de uso nos modelo de casos de uso. Como exemplos de especificação suplementar podem citar: atributos de qualidade do sistema: confiabilidade, usabilidade e etc; sistemas operacionais e etc.

- **Análise e Projeto:** Os objetivos desse fluxo é o de mudar os requisitos em um projeto do que o sistema será; Obter uma arquitetura robusta do sistema e adaptar e adaptar o projeto de acordo com o ambiente de implementação. O modelo de análise e projeto é o principal objetivo dessa disciplina.O modelo de análise e projeto possui a realização dos casos de uso.A realização de casos de uso expressa o que o design espera de um caso de uso,um exemplo seria os diagramas de classes das classes e dos subsistemas participantes.
- **Implementação:** o objetivo é construir um sistema,produzindo o código executável e organizando o código em termos de subsistemas de implementação organizados em camadas;implementar classes e objetos em termos de componentes(arquivo fonte,executável,binários e etc). A disciplina de implementação limita-se ao escopo do sistema e como as classes individuais devem ser testadas em unidades.
- **Teste:** a disciplina de teste oferece serviços a outras disciplinas.O teste foca principalmente na avaliação da qualidade do produto,realizada através de algumas práticas:encontrar e documentar erros na qualidade do software,validar as funções da maneira que foi projetada,verificar se os requisitos foram implementados de forma correta...Uma diferença interessante entre o teste e as outras disciplinas do RUP é que a sua principal finalidade é localizar e expor os pontos fracos do software.A diferença é enquanto as outras disciplinas enfocam na abrangência,o teste enfatiza na deficiência.
- **Implantação:** Descreve as atividades que possibilitem que o software esteja disponibilizado para o usuário final. A ênfase da disciplina de implantação é testar o produto no ambiente de implantação,onde são realizado testes beta,antes de ser entregue ao usuário final.

As disciplinas de fluxo de suporte correspondem a 3,que são elas: ambiente,gerência de configuração e mudanças e gerência de projetos.

- **Ambiente:** A disciplina de ambiente oferece o ambiente de suporte para um projeto.Ao fazer isso, ela também serve de suporte a todas as outras disciplinas.A meta das atividades dessa disciplina é oferecer à organização o ambiente de desenvolvimento de software,processos e ferramentas que dará suporte à equipe de desenvolvimento.
- **Gerência de configuração e mudanças:** O gerenciamento de configuração e mudanças envolve a identificação de itens de configuração,restrições a mudanças nestes itens,verificação de mudanças feitas nos itens e definição e gerenciamento das configurações dos mesmos através do processo de desenvolvimento.Os métodos,processos ferramentas que são usados para fazer esse controle em uma organização pode ser considerada como um sistema CM(..). O sistema CM manuseia as informações importantes sobre o processo de desenvolvimento,a implantação e a manutenção,além de conservar o acervo base de artefatos potencialmente reusáveis,resultando da execução destes processos.
É essencial para o controle de numerosos artefatos produzidos pelas várias pessoas que trabalham em um projeto.O controle ajuda a evitar confusões dispendiosas e garante que os artefatos resultantes não são conflitantes em relação a questões como:atualização simultânea,notificação limitada e múltiplas versões.
- **Gerência de projetos:**O gerenciamento de projetos de software é a arte de balancear os objetivos,gerenciamento de riscos e restrições para entregar,com sucesso,um produto que esteja de acordo com as necessidades dos clientes e usuários.A meta do gerenciamento de projetos do RUP é tornar esta tarefa mais fácil.O propósito do fluxo de gerenciamento de projetos é providenciar um framework para gerenciamento de projetos de software;providenciar guidelines práticas para o planejamento,recrutamento de pessoal,execução e monitoração dos projetos e providenciar um framework para o gerenciamento de riscos.

1.3 OpenUP

1.4.1 Origem do OpenUP

O OpenUP surgiu a partir das diferentes necessidades encontradas em vários projetos. Também existe grande influência da atual dinamicidade nos negócios, as quais refletem diretamente nos requisitos dos softwares. Para

Comment [MA16]: Espaço entre blocos de texto...não deve existir

Comment [MA17]: Mesmo comentário do MA7

tanto valoriza a colaboração entre a equipe e os benefícios aos interessados, ao invés da formalidade e entregáveis desnecessários. O conteúdo fornecido é considerado completo, pois são encontradas definições de papéis, artefatos e processos necessários para o desenvolvimento de um projeto de pequeno porte, com equipe co-localizada.

1.4.2 OpenUp e suas características

O OpenUP (*Open Unified Process*) é um processo de desenvolvimento de software de código aberto, projetado para equipes pequenas e centralizadas, que é atualmente mantido pelo Projeto Eclipse, que define um *framework* de processo de desenvolvimento de *software*. O OpenUp foi inicialmente desenvolvido pela IBM com base nos processos RUP e XP, tendo como principal objetivo reunir as melhores características de cada uma dessas metodologias. Assim, este processo unificado aplica uma abordagem iterativa e incremental dentro de um ciclo de vida estruturado. Contudo, abraça uma filosofia pragmática e ágil que foca na natureza colaborativa do desenvolvimento de *software*. [Monteiro, J.M e Ybanez, M, 2009]

Comment [MA18]: Necessita de espaço antes e depois do título da subseção

Além disso, é um processo independente de ferramenta e de pouca cerimônia que pode ser estendido para direcionar uma grande variedade de tipos de projeto.

Comment [MA19]: Verificar padrão

O OpenUP é um processo: **mínimo**: apenas conteúdos fundamentais do processo são definidos; **completo**: o processo abrange todas as fases do ciclo de vida de desenvolvimento; **extensível**: o processo pode ser utilizado na forma como foi definido, mas permite que novos conteúdos de processos sejam acrescentados para atender de forma mais completa as características de um determinado projeto. [Cunha, C.E. e Vasconcelos, A.M.]

1.4.3 Visão Geral do OpenUp

O processo pode ser facilmente entendido através das 3 camadas abaixo descritas:

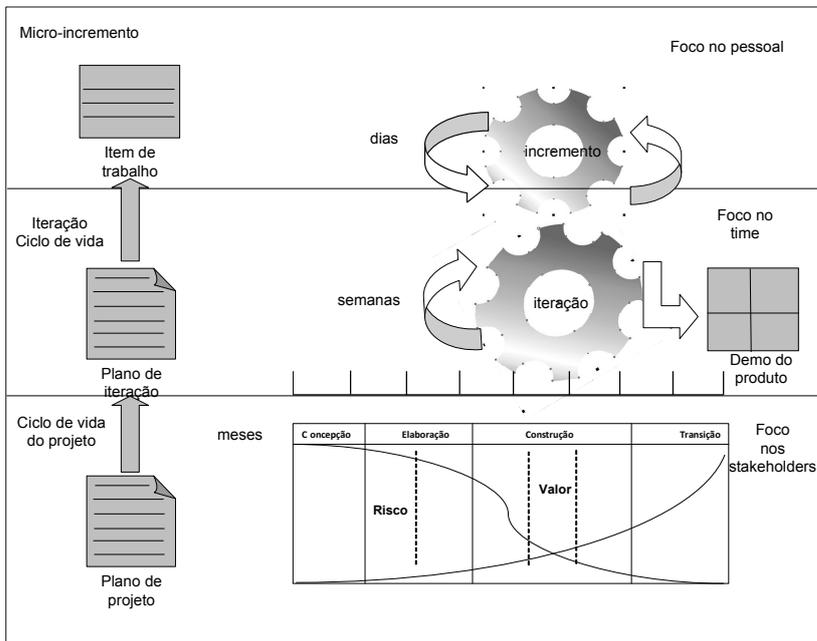


Figura 1.9 Visão Geral do OpenUp

Ciclo de Vida do Projeto – Fases com foco nas necessidades dos Stakeholders.

O OpenUP apresenta a mesma distribuição de fases já conhecidas no RUP, onde o critério de saída de cada fase é no mínimo atender as seguintes respostas:

Iniciação: Todos os stakeholders concordam com o escopo e objetivos do projeto ?

Elaboração: Todos concordam com a arquitetura proposta e o valor entregue ao cliente considerando os riscos levantados?

Construção: Existe uma aplicação que está quase pronta rodando bem próxima a ser finalizada?

Transição: A aplicação está finalizada e o cliente satisfeito?

Comment [MA20]: Quando existir esses espaços gigantes entre palavras, dá um enter no final da frase que corrige

Ciclo de Vida da Iteração com foco no time:

Além da divisão por fases já conhecida, o OpenUP divide o projeto em iterações (também conhecidas como Sprints segundo a metodologia SCRUM) planejadas que podem variar de alguns dias a algumas semanas (a média recomendada é de 4 semanas podendo ser reduzida ou aumentada em até aproximadamente 6 semanas). Ao final de cada iteração deve ser gerado um incremento ao Produto (Build executável ou demo). Ao final de cada iteração geralmente é realizada uma retrospectiva e avaliação onde são discutidas as lições aprendidas e a saúde do projeto. Vale mencionar que o principal objetivo da retrospectiva é aprender com erros e acertos e não apontar culpados.

Micro incrementos com foco individual: Um micro incremento é a execução de um pequeno passo que deve ser mensurável para alcançar os objetivos de uma iteração. Este pode representar o resultado de alguns dias ou horas de trabalho de uma pessoa ou um grupo determinado.

1.4.4 Fases do OpenUp

O OpenUp é dividido em quatro fases: concepção, elaboração, construção e transição. Cada fase pode ter quantas iterações forem necessárias, dependendo do grau de incerteza do domínio de negócio, da tecnologia a ser utilizada, da complexidade arquitetural, do tamanho do projeto, etc. Este processo de entrega define um processo de desenvolvimento de software que suporta os princípios fundamentais do OpenUP. Foi projetado para suportar equipes pequenas e co-localizadas, com de 3 a 6 membros e que trabalhe em um projeto que irá durar de 3 e 6 meses.

1.4.4.1 Concepção

A fase de concepção no OpenUp é responsável por conseguir que se tenha um entendimento simultâneo entre todos os *stakeholders* dos objetivos do ciclo de vida para o projeto.

Há quatro objetivos que clarificam o escopo, os objetivos do projeto e a viabilidade da solução pretendida: **Entender o que construir** a partir da visão dos stakeholders a respeito do produto a ser desenvolvido, quem é o stakeholder do sistema e porque; **Definir** o escopo dos sistemas e seus limites; **Identificar** quais os requisitos mais críticos; Definir pelo menos uma **arquitetura candidata** e sua aplicação prática e **Entender** o custo, cronograma e os riscos associados ao projeto.

1.4.4.2 Elaboração

Com a fase de elaboração o propósito é o de estabelecer uma linha de base da arquitetura do sistema para o volume de esforço de desenvolvimento na próxima fase. Algumas finalidades para a fase de elaboração podem ser citadas, como:

Obter um entendimento mais detalhado dos requisitos permite ao usuário criar um plano mais detalhado e obter comprometimento dos *stakeholder*; Projete, implemente e teste um esqueleto da estrutura do sistema; Apesar da funcionalidade não estar completa ainda, a maior parte das interfaces entre os blocos sendo construídos é implementada e testada. Isto é conhecido como uma arquitetura executável; **Mitigue os riscos essenciais** e produza um cronograma e uma estimativa de custos precisos. Muitos riscos técnicos são resolvidos como resultado do detalhamento dos requisitos e do projeto, implementação e teste da arquitetura e Refine e detalhe o plano de projeto de alto nível.

1.4.4.3 Construção

A finalidade da fase de construção é de terminar o desenvolvimento do sistema baseado na arquitetura colocada na linha de base. Existem objetivos para a fase de Construção que nos ajudam a ter o desenvolvimento com custo eficiente de um produto completo, como por exemplo:

Desenvolver de forma iterativa um produto completo com a descrição dos requisitos que faltam, preencher os detalhes do projeto, terminar a implementação e testar o software; Liberar a primeira versão beta do sistema e determinar se os usuários já estão prontos para que a aplicação possa ser implantada; **Minimizar os custos de desenvolvimento e conseguir algum grau de paralelismo** entre os desenvolvedores ou as equipes de desenvolvimento, como por exemplo, atribuindo os componentes que podem ser desenvolvidos independentemente para desenvolvedores distintos.

1.4.4.4 Transição

A fase de Transição possui alguns objetivos que ajudam a fazer um ajuste na funcionalidade, desempenho e na qualidade total do produto beta proveniente da fase anterior, como por exemplo:

Executar o teste Beta para validar se as expectativas dos usuários foram atendidas; Realizar algumas atividades de ajuste fino, tais como reparação de erros e melhorias no desempenho e na usabilidade; Obter a concordância dos *stakeholders* de que a distribuição está completa, incluindo vários níveis de testes para aceitação do produto, como testes formais, informais e beta; **Melhorar o desempenho de projetos futuros com documentação das lições aprendidas** e Melhorar o ambiente de processos e ferramentas para o projeto.

1.4.5 Princípios básicos do OpenUp

O OpenUP é baseado em 4 princípios básicos que são:

- a) **Equilibrar as prioridades concorrentes para maximizar o valor para os stakeholders** - para que se alcance o sucesso, *stakeholders* e participantes do projeto devem convergir para um claro entendimento e concordar com três fatores: Problema a ser resolvido, Restrições impostas à equipe de desenvolvimento (custo, cronograma, recursos, regulamentos) e Restrições impostas à solução. Coletivamente, estes três itens representam os requisitos para o desenvolvimento do sistema. O desafio de todos os participantes do projeto é criar uma solução que maximize o seu valor para os *stakeholders*, mesmo que sujeitos a restrições. O equilíbrio está em fazer uma análise crítica do custo-benefício entre características desejadas e as decisões de projeto subsequentes que definem a arquitetura do sistema.

b) Focar na evidenciação da arquitetura

Sem uma base arquitetural, um sistema não evoluirá de forma eficiente, será difícil organizar a equipe ou comunicar idéias sem foco técnico comum que a arquitetura fornece. Dessa forma, utilize a arquitetura como um ponto focal, para que desenvolvedores possam alinhar seus interesses e idéias ao se tornar evidentes as decisões técnicas essenciais tomadas em cada evolução arquitetural.

c) Colaborar para alinhar os interesses e compartilhar o entendimento

Em uma equipe cada membro tem seus próprios conhecimentos, habilidades e maneiras de fazer as coisas. Este princípio tem a finalidade de alinhar essas diferenças de forma que o projeto seja beneficiado bem como fazer com que todos os membros da equipe tenham um entendimento sobre o projeto. O contínuo aprendizado também é estimulado por este princípio fazendo com que cada membro da equipe desenvolva mais habilidades e incremente seus conhecimentos.

d) Evoluir para continuamente obter feedback e promover melhorias

O objetivo deste princípio é fazer com que através de feedbacks, obtenha-se modos de melhorar o produto e também o processo da equipe envolvida. Através de feedbacks, pode-se identificar potenciais riscos e tratá-los mais cedo durante o projeto.

É importante ter o objetivo do projeto de maneira clara ao próprio entendimento para que seja possível fazer uma medição do progresso e identificar possíveis melhorias no processo.

1.4.6 Disciplinas

As disciplinas do OpenUP são divididas em 6. Elas serão detalhadas logo abaixo:

a) Análise e Projeto

Os propósitos da Análise e Projeto são: transformar os requisitos em um projeto do que será o sistema, desenvolver uma arquitetura robusta para o sistema e adaptar o projeto para corresponder com ambiente de implementação. A disciplina de Análise e Projeto está relacionada a outras disciplinas, como por exemplo: a disciplina de Requisitos provê a primeira entrada para a Análise e Projeto; a disciplina de Implementação implementa o projeto; a disciplina de Teste testa o projeto do sistema durante a Análise e Projeto e a disciplina de Gestão de Projetos planeja o projeto e cada iteração.

b) Gerência de Configuração e Mudança

As finalidades desta disciplina é: Manter um conjunto de produtos de trabalho consistente a medida que evolui, manter construções de software consistentes, fornecer meios eficientes para se adaptar às mudanças, replanejando o trabalho adequadamente, fornecer dados para a medição do progresso e está relacionada ao controle de mudanças de artefatos pela configuração e a habilidade de manter versões e configurações consistentes dos artefatos.

c) Implementação

A disciplina de implementação tem como propósitos: Construir o sistema de forma incremental e verificar que as unidades técnicas usadas para construir o sistema funcionem como especificado. Esta disciplina se relaciona com as outras disciplinas da seguinte forma: A disciplina de Requisitos define o que será implementado; a de Análise e Projeto organiza e define o escopo da implementação; a de Teste valida que a construção do sistema atende aos requisitos; a de Gerência de Configuração e Mudança fornece mecanismos para controlar mudanças no sistema em construção e a disciplina de Gerência de Projeto planeja quais funcionalidades serão implementadas em cada iteração.

d) Gerência de Projetos

A disciplina de Gerência de Projetos está focada em: manter a equipe focalizada na entrega contínua do produto de software testado para a avaliação dos *Stakeholders*; ajudar a priorizar a sequência de trabalho; ajudar a criar um ambiente de trabalho eficaz para maximizar a produtividade da equipe; manter os *Stakeholders* e a equipe informados sobre o progresso do projeto e fornecer uma estrutura para controlar o risco do projeto e para adaptar-se continuamente às mudanças

O gerenciamento de projeto age como um elo entre os *Stakeholders* e a equipe de desenvolvimento. É interessante que as atividades da gerenciamento de projeto adicionem valor ao criar um ambiente de trabalho de elevado desempenho onde os *Stakeholders* tenham confiança na habilidade da equipe de conhecer as capacidades e restrições da plataforma técnica e de entregar com sucesso algo valioso e que os membros da equipe de projeto entendam as necessidades dos *Stakeholders*, produzindo continuamente um produto de software para avaliação.

e) Requisitos

A disciplina de requisitos tem como atividades: entender o problema a ser resolvido; entender as necessidades dos *Stakeholders*; definir os requisitos para a solução; definir o escopo do sistema; identificar interfaces externas ao sistema; identificar restrições técnicas na solução; fornecer a base para o planejamento das iterações e fornecer a base inicial para a estimativa de custo e cronograma. Para realizar essas atividades, é importante que compreendam a definição e o escopo do problema que estamos tentando resolver. A disciplina de Requisitos é relacionada às outras disciplinas das seguintes maneiras:

A disciplina de Análise e Projeto obtém suas entradas primárias a partir da disciplina de Requisitos; a de Teste valida o sistema de acordo com os requisitos; a de Gerência de Configuração e Mudança fornece os mecanismos para controlar as mudanças nos requisitos; a de Gerência de Projeto planeja o projeto e atribui os requisitos a cada iteração analisando os requisitos priorizados e atribuindo o trabalho.

f) Teste

O propósito desta disciplina é: Encontrar e documentar defeitos, validar e provar as suposições feitas no projeto e requisitos especificados através de demonstrações concretas, validar que o produto de software foi feito como projetado e validar que os requisitos estão apropriadamente implementados.

A disciplina de Teste está relacionada às outras disciplinas das seguintes maneiras: A disciplina de Requisitos captura requisitos para o produto de software, que é um das contribuições primárias para identificar que testes executar; A de Análise e Projeto determina o projeto apropriado para o produto de software, que é outra contribuição importante por identificar que testes executar; A de Implementação produz construções do produto de software que é validado pela disciplina de Teste; A de Gerência de Projeto planeja o projeto e o trabalho necessário em cada iteração e a de Gerência de Configuração e Mudança controla mudanças dentro do projeto. O esforço de teste verifica se cada mudança foi completada adequadamente. Ativos de teste são mantidos abaixo da gerência de configuração.

1.5 MSF

1.5.1 Origem do MSF

O Microsoft Solutions Framework(MSF) surgiu em 1994, a partir da análise de como a Microsoft desenvolve seus produtos. Basicamente o MSF é uma compilação das boas práticas utilizadas pela empresa, que foi criado tanto para uso interno como para uso de seus clientes. Porém, apesar de ter sido criado pela Microsoft, o MSF aborda basicamente o processo de construção de soluções, não se prendendo ao uso de produtos desta empresa.[Cardim.I.C.,2006]. Inicialmente,surgiu o MSF 3.0 que era composto por quatro elementos básicos: princípios fundamentais,modelo de processo,modelo de equipe e disciplinas. Dentro da totalidade da popularização dos processos ágeis,a versão 4.0 do MSF foi lançada para atender duas extremidades específicas que são: MSF Agile Software Development(MSF4ASD) abordando processos ágeis e na outra vertente o MSF for CMMI Process Improvement(MSF4CMMI) que aborda os processos tradicionais. No entanto,mesmo assim,existe muita semelhança entre as duas versões.

Ao longo do tempo o MSF já se chamou: MDF (Microsoft Development Framework) e PCM (Product Cycle Model). Consolidando-se como MSF após o livro Microsoft Secrets de Michael Cusumano (1995). A história da própria Microsoft e do MSF se misturam, podemos então considerar o MSF como a “atitude mental da Microsoft” para seu modelo de negócios.

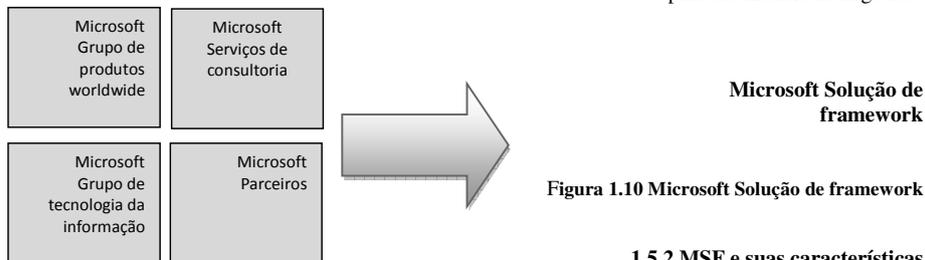


Figura 1.10 Microsoft Solução de framework

1.5.2 MSF e suas características

O MSF é um processo iterativo e incremental,onde as atividades são realizadas de forma cíclica, resultando em produtos incrementais;assim como o RUP e OpenUp. O MSF permite uma fácil compreensão tanto por parte da equipe como do cliente,além de ser bem flexível em sua aplicação; Há o aprendizado contínuo do projeto,onde cada lição é uma grande lição para equipe envolvida no processo de desenvolvimento; Pode-se perceber a evolução da curva de produtividade; Acompanhar a autonomia técnica crescente dos colaboradores a cada dia dentro do projeto e Possui uma teoria de controle,onde pequenas iterações permitem calibrar com mais precisão e agilidade estimativas e reduzir margens de erro.

1.5.3 Visão Geral do MSF

Um projeto MSF é regido por iterações ou ciclos. A cada ciclo, cada componente da equipe executa suas funções e atualiza o resultado do seu trabalho conforme a necessidade. Os ciclos se repetem até que o projeto seja concluído ou cada versão seja lançada. Cada componente da equipe será responsável por um ou mais papéis, dependendo do tamanho ou da complexidade do projeto. O MSF é dividido em dois modelos: Modelo de time(Team Model) e seus respectivos processos e Modelo de processo(Process Model).

O modelo de time habilita a escalabilidade do projeto,identifica quem vai trabalhar durante o projeto e definir cada time com um responsável.

O modelo de processo será falado na próxima seção.Este é um modelo que trabalha em conjunto com o modelo de time organizando o processo em fases distintas.

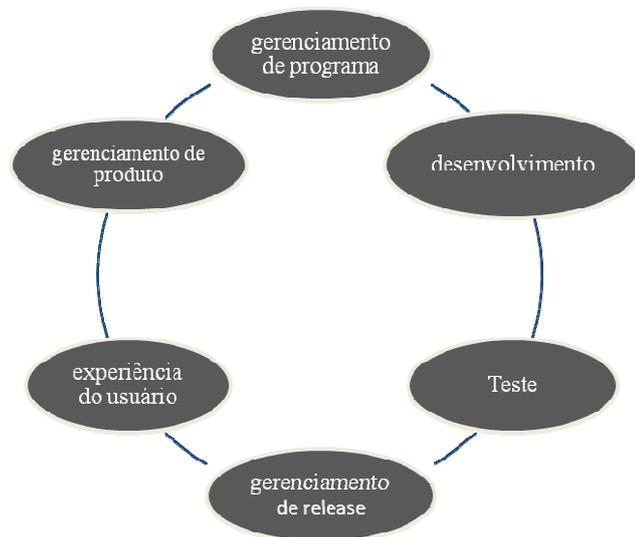


Fig 1.11. Modelo de times

O modelo de time conjuntamente com o modelo de processos possuem objetivos e metas da equipe. A fase de **Gerenciamento de programa** atua nas áreas de gerenciamento de projeto, soluções em arquitetura, garantia de processos e serviços administrativos; **Desenvolvimento** atua nas áreas de consultoria tecnológica, implementação da arquitetura e design, desenvolvimento da aplicação e desenvolvimento da infra-estrutura; **Teste** atua nas áreas de plano de teste, engenharia de teste e reporte de teste; **Gerenciamento de release** está relacionada aos processos de infra-estrutura, suporte, operações, logística e gerenciamento comercial das publicações; **Experiência do usuário** atua nas áreas de acessibilidade, internacionalização, treinamento/material e suporte, pesquisa de usabilidade e teste, advogado do usuário e design de interface; **Gerenciamento do produto** satisfaz os clientes com valor de negócio, marketing, advogado do cliente e planejamento do produto. O princípio básico deste modelo é que cada um desses papéis aborda um objetivo importante para o projeto. Por isso todos os papéis devem estar representados e poder comunicar-se entre si, além de participar das decisões de projeto.

1.5.4 Fases do MSF



1.12. Fases do MSF

O modelo de processos do MSF é dividido em 5 fases: Previsão, Planejamento, Desenvolvimento, Estabilização e implantação. Cada fase descreve um conjunto de subprodutos que devem ser entregues, assim como marcos que devem ser atingidos e os respectivos critérios de aceitação. Essas fases serão descritas a seguir:

- a) **Previsão:** Esta fase tem como foco fazer com que a equipe tenha uma visão comum do projeto. As atividades realizadas nessa fase são: formação de equipe, elaboração do projeto (visão mais ampla do que o projeto deve fazer) e a definição do escopo do projeto.
- b) **Planejamento:** Durante essa fase a equipe estima custos, prepara plano de trabalho e programa os entregáveis. É nesta fase que a equipe lista os requisitos do projeto (requisitos de negócio, do usuário, operacionais e de sistema). Esses são os requisitos que o MSF reconhece.
- c) **Desenvolvimento:** nessa fase a equipe implementa a maioria dos componentes da solução. Os principais artefatos gerados nessa fase são: código fonte e executável, scripts de instalação e configuração, especificação funcional, elementos de suporte a performance, especificação e casos de teste.
- d) **Estabilização:** esta fase tem como objetivo testar o que foi implementado na fase anterior. Esses testes enfatizam o uso do sistema simulando o ambiente de funcionamento. A equipe se preocupa em resolver erros encontrados nos testes e prepara a solução para liberação. Esta fase só termina quando todos os erros são corrigidos.
- e) **Implantação:** nessa fase, a equipe estabiliza o produto e obtém a aprovação do cliente final. As atividades de estabilização do produto podem continuar durante este período enquanto os componentes do projeto são transferidos do ambiente de teste para o ambiente de produção. O aspecto relacionado a satisfação do cliente deve ser coletado em todas as fases.

1.5.5 Princípios básicos MSF

Nessa seção, iremos apresentar os princípios básicos do MSF, que se compõem na filosofia e comportamento para as equipes que utilizam o conceito do MSF no desenvolvimento dos seus projetos de software.

a) **Parceria com o cliente**

A ideia é ter o cliente como membro do time para fazer validações constantes. Dessa forma está comprometido com o projeto e possa entender o que está sendo feito. Assim, os riscos do projeto podem ser mitigados.

b) Qualidade é trabalho de todos

O termo qualidade é muito subjetivo. Promover qualidade significa investir em pessoas, ferramentas e processos. Requer tanto prevenção de “bugs/problemas” quanto verificação de possíveis soluções.

c) Trabalho em direção a uma visão compartilhada

Segundo Steve McConnell, uma pesquisa feita com 75 times mostra que em todos os casos em que o time funciona eficazmente, todos os membros conheciam os seus objetivos! Através da visão do projeto os membros do time podem definir prioridades, tomar decisões e garantir que os esforços estejam alinhados aos resultados que se esperam.

d) Manter-se ágil, adaptar-se às mudanças

Quanto mais uma organização procura maximizar o impacto no negócio de um investimento em tecnologia, mais ela descobre novos ambientes e desafios.

Os projetos de tecnologia tem uma característica relevante: mudanças constantes. Alterações no projeto devem ser esperadas e é impossível isolar a entrega do projeto. Com isso, o MSF foi desenvolvido para gerenciar e antecipar a mudanças. Todas as alterações são aprovadas pela equipe, dessa forma melhora o impacto das mudanças e mitiga os impactos negativos.

e) Encorajar comunicação aberta

Para desenvolver um bom trabalho, é necessário que todos os membros da equipe tenham conhecimento prévio do que está sendo feito. Isso minimiza o desconhecimento, incertezas e retrabalho.

f) Autorização dos membros da equipe

Dar poder aos membros da equipe é um grande diferencial do MSF, pelo fato de pregar um modelo em rede hierárquica, onde cada membro é responsável pela entrega do produto.

g) Estabelecer a responsabilidade desobstruída e responsabilidade compartilhada

A definição clara dos papéis e das responsabilidades de cada membro da equipe é um dos principais fatores de sucesso. Sem isso implica em um retrabalho duplicado e certa insegurança em relação a função. Um estudo mostrou que esse princípio diminui as incertezas quanto “o que”, “quem”, “quando” e “por que” com os resultados, tornando o trabalho mais eficiente e compensador.

h) Foco em entregar um valor de negócio

Os projetos de tecnologia não devem focar em “entregas de tecnologia”, mas em “entregas com valor tangível ao negócio”. O projeto tem que possuir uma ligação íntima com o negócio, se não existir essa ligação pode resultar em entregas com atraso e projetos cancelados.

i) Aprender com todas as experiências

Estatísticas mostram repetições de falhas em projetos. De fato, isso mostra que as pessoas não estão aprendendo com os erros para mudar esse quadro. E esse quadro piora mais ainda por que estamos diante prazos curtos e recursos limitados. O MSF recomenda revisões, coleta de lições aprendidas no projeto e criação de um documento no final do projeto, para que os erros não repitam.

j) Criar sempre possibilidade de serem entregues produtos

O time deve crer que o produto deve estar pronto para ser entregue a qualquer momento, mesmo no contexto de desenvolvimento de soluções.

1.5.6 Disciplinas do MSF

As disciplinas são necessárias durante o ciclo de vida dos projetos e são guias constantes para cada modelo. O MSF assume três disciplinas que são:

- a) **Gerenciamento de projeto:** é uma disciplina que incorpora atividades de diversas áreas de conhecimento; a maioria das responsabilidades sabidas da área de “gerência de projeto” são atribuídas ao indivíduo responsável pelo papel de gerente de projeto. A disciplina de gerenciamento de projeto ajuda o time a obter sucesso sem perder performance com recursos adicionais que não fornece valor suficiente aos recursos investidos.
- b) **Gerenciamento de risco:** é o gerenciamento pró-ativo, compreensivo, visando o sucesso e diminuindo fatores negativos que impactariam no fracasso do projeto. A gerência de riscos é uma resposta à incerteza intrínseca em projetos de tecnologia.
- c) **Gerenciamento de Aprendizado:** A disciplina de gerenciamento de aprendizado identifica habilidades exigidas pelo time, alocando desse modo, recursos que o projeto necessita e criando oportunidades de aprendizado e crescimento.

Considerações finais????

Comment [MA21]: Considerações finais????

1.6 Tópicos de Pesquisa

Se quiser saber mais sobre o processo RUP e quais empresas o adotou, acesse esses sites:

<http://groupware.les.inf.puc-rio.br/groupware/publicacoes/2008.SBSI>

[Pimentel.RUP3CGroupware.pdf](#)

http://www.oremi.com.br/artigo/arquivos/080715130256_2007WOSES.pdf

<http://www.cingo.com.br/>

Comment [MA22]: Precisa de mais tópicos de pesquisa

1.7 Sugestão de Leitura

Durante a escrita do capítulo, foram identificados algumas sugestões de leitura relacionadas ao processo RUP. Como por exemplo:

- O livro Introdução ao RUP: Rational Unified Process do autor Phillippe Kruchten;
- O livro Conheça o Rational Unified Process do autor Mauro Viana
- Rational Unified Process: uma abordagem gerencial disponível em: http://www.de9.ime.eb.br/~tssouza/eng_soft/Trabalho%20RUP/Mono_RUP.pdf
- http://inf.unisul.br/~vera/egs/Rup_iso9000.pdf
- http://www.wthreex.com/rup/process/workflow/requirem/co_req.htm

Comment [MA23]: Deve contemplar os outros processos também

1.8 Exercícios

1. Por que o RUP é considerado um processo tradicional? Justifique a sua afirmação. Com base em leituras complementares, cite alguns exemplos de empresas que utilizam o RUP.
2. O que faz do processo RUP um modelo iterativo e incremental?

3. As fases do RUP são definidas em 4. Fale de forma resumida sobre os objetivos primordiais de cada fase.
4. Quais são as disciplinas do processo Rup? Qual o papel de cada uma?
5. Quais são as diferenças relevantes do RUP comparado ao OpenUp?
6. Fale sobre os princípios básicos do OpenUp.
7. Quais são as fases do MSF? Detalhe cada uma.
8. Fale um pouco sobre o modelo de time do MSF.
9. Explique os princípios do MSF. Selecione os que você acha mais importantes para sua empresa.

1.9 Referências Bibliográficas

<http://www.wthree.com/rup/portugues/index.htm>

<http://www.linhadecodigo.com.br/Artigo.aspx?id=79>

<http://javafree.uol.com.br/artigo/871455/Obtendo-Qualidade-de-Software-com-o-RUP.html>

<http://www-01.ibm.com/software/rational/>

<http://www.linhadecodigo.com.br/Artigo.aspx?id=1471>

<http://blogs.msdn.com/bgroth/archive/2005/03/08/389839.aspx>

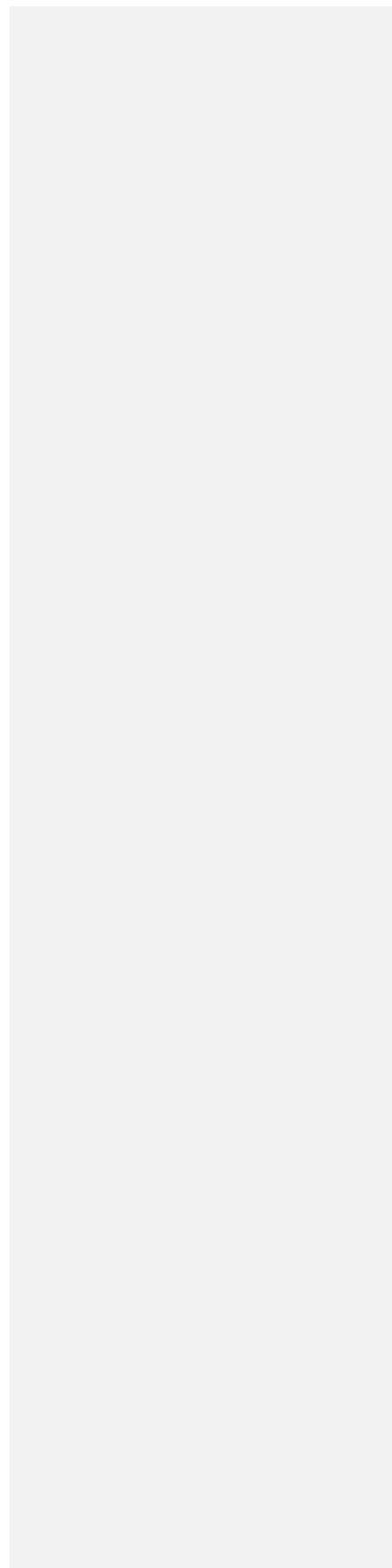
<http://www.devmedia.com.br/articles/viewcomp.asp?comp=4574>

Comment [MA24]: Formatar as referências

<http://www.laps.ufpa.br/yomara/paginas2/aps/processo%20unificado%20rup.pdf>
<http://www.cci.unama.br/margalho/portaltcc/tcc2003/d2615.pdf>
<http://guaiba.ulbra.tche.br/pesquisas/2008/artigos/sistemas/328.pdf>
<http://www.ime.uerj.br/~vera/projeto/apostila.pdf>
[http://imasters.uol.com.br/noticia/1861/gerencia/modelos de ciclo de vida por que precisamos deles no desenvolvimento/](http://imasters.uol.com.br/noticia/1861/gerencia/modelos%20de%20ciclo%20de%20vida%20por%20que%20precisamos%20deles%20no%20desenvolvimento/)
<http://www.cordeiro.pro.br/aulas/engenharia/processoDeSoftware/ciclos.pdf>
<http://www-01.ibm.com/software/awdtools/rup/>
<http://www.cesumar.br/pesquisa/periodicos/index.php/iccesumar/article/viewFile/133/71>
<http://www.eclipse.org/epf/general/OpenUP.pdf>
<http://www.infobrasil.inf.br/iConstructor/Custom/anais2009/Integrando%20Metodologias%20%20C3%81%20geis%20e%20Modelos%20de%20Maturidade%20de%20Software%20Um%20Estudo%20de%20Caso.pdf>
<http://www.ibm.com/developerworks/rational/library/oct05/kroll/>
<http://www.cin.ufpe.br/~tg/2005-2/icc2.pdf>
http://dsc.upe.br/~tcc/20062/Monografia_TiagoMoraes.pdf

Dúvidas:
Como é vc que vai abrir o livro... no que diz respeito a processos e talz... abordagens como processo ciclo de vida e outras mais serão explicadas?
Tipo, quando for falar sobre processos em algum momento dizer o que é um processo, pode ser frase de um autor e fazer referência e por ai vai.

Índice do Capítulo



<u>2.1 INTRODUÇÃO A PROCESSOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE.....</u>	<u>103</u>
<u>2.2 O MANIFESTO ÁGIL</u>	<u>104</u>
<u>2.3 PRINCIPAIS PROCESSOS ÁGEIS</u>	<u>105</u>
<u>2.4 EXTREME PROGRAMMING (XP)</u>	<u>106</u>
2.4.1 VALORES DO XP	106
2.4.2 PRINCÍPIOS DO XP	107
2.4.3 PRÁTICAS DO XP	109
2.4.4 PAPÉIS DO XP	112
2.4.5 CICLO DE VIDA DO PROJETO XP	113
<u>2.6. SCRUM.....</u>	<u>115</u>
2.6.1 CARACTERÍSTICAS DO SCRUM.....	115
3.6.2 PAPÉIS DO SCRUM	115
2.6.3 PRÁTICAS DO SCRUM	116
2.6.4 CICLO DE VIDA DO SCRUM	117
<u>2.6 FEATURE DRIVEN DEVELOPMENT.....</u>	<u>117</u>
2.6.1 CARACTERÍSTICAS DO FDD	117
2.6.2 PAPÉIS DO FDD.....	118
2.6.3 PRÁTICAS DO FDD	119
2.6.7 CICLO DE VIDA DO FDD.....	120
<u>2.7. TÓPICOS DE PESQUISA.....</u>	<u>122</u>
<u>2.8. SUGESTÕES DE LEITURA.....</u>	<u>122</u>
<u>2.9. EXERCÍCIOS</u>	<u>123</u>
<u>REFERÊNCIAS</u>	<u>123</u>

Capítulo

2

Processos Ágeis de Desenvolvimento de Software

Márcio Amorim de Medeiros, Milton Moura Campos Neto

Este capítulo apresenta uma nova abordagem de desenvolvimento de software, os Processos Ágeis, que surgiram para reduzir os problemas e custos em comparação aos Processos Tradicionais. Ao longo deste capítulo contextualizaremos a respeito do paradigma ágil e utilizaremos, também, terminologias como métodos e metodologias nessa abordagem, com ênfase no *Extreme Programming (XP)*, *Scrum* e *Feature Driven Development (FDD)*.

2.1 Introdução a Processos Ágeis de Desenvolvimento de Software

Os processos tradicionais de desenvolvimento de software geralmente não se adequam à realidade de algumas organizações, em especial, as pequenas e médias fábricas de software que não possuem recursos para seguirem processo algum. Os processos ágeis surgiram como uma nova tendência de desenvolvimento para melhorar a qualidade dos sistemas e reduzir a quantidade de projetos fracassados, eliminando gastos com documentação excessiva, enfatizando a comunicação, mais flexíveis à mudança e privilegiando as atividades que agregam valor ao negócio.

Tanto os métodos pesados, quanto os ágeis possuem o mesmo objetivo, satisfazer as necessidades dos usuários construindo sistemas de qualidade. A diferença entre eles está nos princípios utilizados por cada um.

[SATO 2007] Os princípios relacionados aos processos tradicionais são citados no **Capítulo 1**, já os ágeis são detalhados na seção **O Manifesto Ágil** deste capítulo.

Atualmente, mudança é algo bastante comum na vida de um software, a fim de garantir adaptação do sistema às novas necessidades do cliente, instituições ou do mercado. Os processos tradicionais tendem a tentar planejar grande parte do software por um longo período antes de iniciar a implementação. Com isso, o software demora a ser disponibilizado ao cliente. Durante esse tempo pode surgir novos padrões, políticas e tecnologias que afetam os requisitos do software, o cliente pode perceber que alguma funcionalidade não está conforme solicitado ou precisa de outras. Esses fatores implicam em mudança no sistema, que não é bem-vinda nos métodos tradicionais, pois a fase de planejamento já foi concluída.

Outro fator comum no desenvolvimento tradicional é a implementação de funcionalidades que não agregará valor ao cliente, ou seja, o sistema vai disponibilizar funcionalidades aos usuários que serão pouca ou nunca utilizada, enquanto outras funções mais prioritárias ainda não foram implementadas.

As metodologias ágeis surgiram com a finalidade de desburocratizar o processo de desenvolvimento. Elas tentam se adaptar e fortalecer com as mudanças, até mesmo a ponto de se auto-modificarem. Os clientes têm, em curto espaço de tempo, versões de software executável, onde são priorizadas as funcionalidades que agregam mais valor ao seu negócio. Com isso, ele já pode sugerir novas funcionalidades e correções.

Na agilidade, outro fator determinante é o fato de “não documentar, apenas por documentar”. Só é documentado aquilo que for necessário em outro momento e que justifique o esforço e recursos gastos na documentação. Segundo [Beck, 2000], nos processos ágeis, a documentação se restringe às histórias dos usuários.

Comment [A25]: SUGESTÃO: “Métodos tradicionais e ágeis possuem o mesmo objetivo...”

Comment [A26]: SUGESTÃO: “Já foram abordados no Capítulo 1”

descrições do funcionamento do sistema, baseada em uma simples história metafórica, que é usada para ajudar envolvidos no projeto a ter uma visão de seu funcionamento e entender os elementos básicos do projeto e seus relacionamentos.

Vale ressaltar que os Processos ágeis são orientados a pessoas ao invés dos tradicionais que são orientados a processos. Metodologias ágeis afirmam que nenhum processo jamais será semelhante à habilidade da equipe de desenvolvimento. Em vista disso, o papel do processo é dar suporte à equipe e seu trabalho. Nas próximas seções será detalhado desde o Manifesto Ágil, onde serão citados os princípios comuns aos métodos ágeis existentes, e citadas e caracterizadas as principais metodologias.

Comment [A27]: Pouco vago, como é introdução sugiro retirar o termo "história metafórica", pois deve ser explicado posteriormente ao falar das práticas.

Comment [A28]: Sugestão: Retirar o termo "Vale ressaltar que"

2.2 O Manifesto Ágil

No início de 2001, 17 especialistas em software reuniram-se para propor um conjunto de princípios e valores para agilizar o desenvolvimento dos seus sistemas tendo como base suas elevadas experiências em programação. Foram motivados pela conclusão de que os processos de desenvolvimento estavam tornando-se cada vez mais longos, atolando as equipes de construção de softwares.

A essência desse movimento é a definição de novo enfoque de desenvolvimento de software, calcado na agilidade, na flexibilidade, nas habilidades de comunicação e na capacidade de oferecer novos produtos e serviços de valor ao mercado, em curtos períodos de tempo. [HIGHSMITH, 2004]

Esse movimento, marco inicial do desenvolvimento ágil de software, foi descrito abaixo:

"Nós estamos descobrindo melhores maneiras melhores de **desenvolver software, fazendo e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar** [AGILE MANIFESTO 2009]:

Indivíduos e interação entre eles mais que processos e ferramentas
Software em funcionamento mais que documentação abrangente
Colaboração com o cliente mais que negociação de contratos
Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

Assinaram esse manifesto: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas.

Os envolvidos se denominaram de Aliança Ágil. Esta abordagem tentava manter a qualidade dos projetos de software permitindo aos mesmos que mudanças fossem inseridas em seus desenvolvimentos, mas que reduzisse seus impactos, esta flexibilidade foi traduzida nos quatro valores vistos acima e em doze princípios que estão mostrados a seguir:

- **Nossa maior prioridade** é satisfazer o cliente através de entregas antecipadas e contínuas de software de valor ao cliente;
- Mudanças de requisitos são bem vindas, mesmo que tardiamente no desenvolvimento. Processos ágeis se aproveitam da mudança para vantagem competitiva do cliente;
- Entrega freqüente de software funcionando, de duas semanas de trabalho até dois meses, com preferência à escala de tempo mais curta;
- Pessoas de software e negócios devem trabalhar juntas diariamente durante o desenvolvimento do projeto;
- **Construa projetos em torno de indivíduos** motivados. **Dê-los o ambiente e o suporte necessário e acredite** neles para fazer o trabalho;
- O Método mais eficiente e efetivo de repassar a informação dentro de uma equipe de desenvolvimento é a conversa face a face
- Software funcionando é a primeira medida de progresso;
- Processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um passo sustentável indefinidamente;

Comment [A29]: Acho que poderia colocar como uma tabela ou quadro e informar a fonte.

Comment [A30]: Reformular

Comment [A31]: Substituir por: "A maior prioridade é..."

Comment [A32]: Substituir por: "Projetos construídos em torno de ..."

- Atenção contínua a excelência técnica e um bom projeto melhoram a agilidade;
- Simplicidade: a arte de maximizar a quantidade de trabalho não feito, é essencial;
- As melhores arquiteturas, requisitos, e projetos emergem de times auto-organizáveis;
- Em intervalos regulares, o time deve refletir sobre como se tornar mais efetivo, então melhora e ajusta seu comportamento de acordo com a reflexão.

2.3 Principais Processos Ágeis

São considerados Processos Ágeis de Desenvolvimento de Software as metodologias que seguem os princípios do Manifesto Ágil – como entrega freqüente, flexão a mudança, foco em pessoas e simplicidade. Os métodos ágeis ganham mais adeptos à medida que estão evoluindo com as novas técnicas e adaptações. A 3ª pesquisa sobre o estado do Desenvolvimento Ágil promovido pela VersionOne [VERSIONONE 2008], e aplicada a mais de três mil entrevistados de 80 países, revela entre diversos indicadores, quais as metodologias ágeis mais utilizadas nas empresas, conforme Figura 3.1. As principais metodologias ágeis serão caracterizadas a seguir:

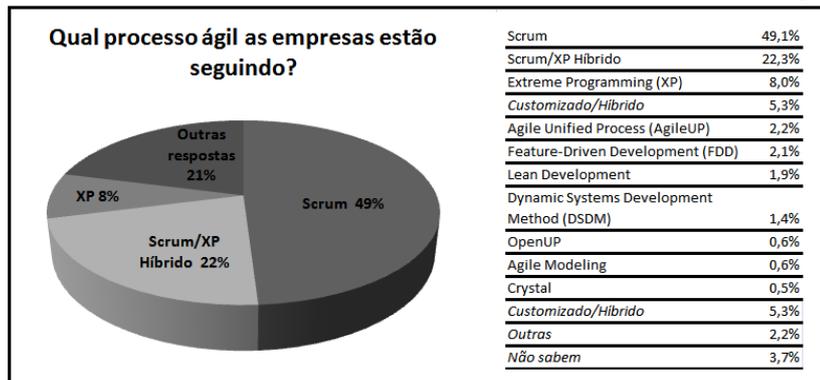


Figura 2.1. Metodologias Ágeis mais utilizadas nas empresas de acordo com a 3ª Pesquisa Anual sobre o Estado do Desenvolvimento Ágil – Ano 2008. Adaptado de [VERSIONONE 2008].

A metodologia *Agile Unified Process* (AUP), também conhecida como AgileUP e desenvolvida por Ambler [AMBLER 2002], é uma versão simplificada do *Rational Unified Process* (RUP), que está detalhado no Capítulo 1.

Ela descreve como desenvolver sistemas utilizando técnicas ágeis, como *Test Driven Development* (TDD), *Agile Modeling* e *Refactoring* no banco de dados para melhorar a produtividade, mesmo assim mantendo-se fiel ao RUP.

A *Lean Development* tem suas raízes na indústria automotiva, ele é uma adaptação para software do *Lean Manufacturing* do revolucionário Sistema Toyota de Produção. Inicialmente proposto por Bob Charette, tem como principais objetivos tentar reduzir em um terço o prazo, o custo e o nível de defeito no desenvolvimento de software e para isso exige um grande comprometimento da alta administração com predisposição inclusive para mudanças radicais.

Baseado no Desenvolvimento Rápido de Aplicações (RAD) e no modelo iterativo e incremental, a metodologia *Dynamic Systems Development Method* (DSDM) se tornou o framework para RAD [STAPLETON, 2003]. A idéia central do DSDM baseia-se no seguinte: ao invés de fixar o escopo de funcionalidades do produto e a partir daí estimar tempo e recursos para alcançar o escopo definido, é preferível fixar tempo e recursos e ajustar o escopo de acordo com estas limitações [COHEN et al., 2003].

Comment [A33]: Seria em itálico?

O *Crystal* não é apenas uma única metodologia, e sim, uma família de métodos denominada, portanto, de Família *Crystal*, proposto por Alistair Cockburn [COCKBURN, 2002]. É categorizada por cores de acordo com a quantidade de pessoas envolvidas. A versão mais ágil e mais documentada é a *Crystal Clear* que pode ser usada em projetos de até oito pessoas, seguida pela *Crystal Yellow* (para times de 8 a 20 pessoas), *Crystal Orange* (para times de 20 até 50 pessoas), *Crystal Red* (para times de 50 até 100 pessoas), e assim por diante.

Já a *Agile Modeling* (AM), apesar de ser citada na pesquisa, não é uma metodologia de desenvolvimento de software. É uma metodologia de modelagem ágil, isto é, AM visa construir e manter modelos de sistemas de maneira eficaz e eficiente, que pode ser utilizada dentro de metodologias ágeis ou nos processos tradicionais, como o RUP.

Os métodos ágeis que se destacam no mercado é o *eXtreme Programming* (XP), que se atém ao desenvolvimento propondo um conjunto de técnicas, e o *Scrum*, que enfatiza o planejamento e gerenciamento dos projetos. Ambos serão tratados com maior detalhe nas próximas seções, além da metodologia FDD, pois é uma boa opção para grandes projetos e para empresas que possuem dificuldades para migrar para um ambiente completamente ágil.

2.4 Extreme Programming (XP)

O *Extreme Programming* (XP) é um processo de desenvolvimento de software ágil, criado oficialmente em 1999 por Kent Beck e Ward Cunningham, quando do lançamento do livro *Extreme Programming Explained* que formalizou e difundiu o referido processo. As experiências dos dois, que culminou no XP, vêm desde o início da mesma década com *Smalltalk*.

A metodologia XP é definida como leve e apoiada em valores, princípios e práticas, cujo foco é o desenvolvimento de um produto que venha atender aos objetivos do cliente e forte adaptabilidade a requisitos não totalmente definidos e em constante mudança [BECK 2004].

O termo “*Extreme*” referencia o emprego de forma extrema das boas práticas da engenharia de *software*, além de suas práticas peculiares que diferencia XP de outras ágeis, como a programação em pares, forte cultura de testes, propriedade coletiva de código entre outras que serão abordadas adiante.

Alguns adeptos do XP o definem como sendo a prática e a perseguição da mais clara simplicidade, aplicado ao desenvolvimento de software [TELLES 2004].

Segundo Kent Beck [2004] o XP inclui em seu arcabouço: a) uma filosofia de desenvolvimento de *software* baseada nos valores (comunicação, *feedback*, simplicidade, coragem e respeito); b) um conjunto de práticas, que expressam os valores, comprovadamente úteis para melhorar o desenvolvimento de software; um conjunto complementar de princípios, técnicas intelectuais que auxiliam a tradução dos valores em práticas, úteis quando as práticas existentes não resolvem se problema particular; e uma comunidade que compartilha dos mesmos princípios e muitas das mesmas práticas.

Nas seções posteriores detalharemos esses valores, princípios e práticas do XP, com base na nova abordagem feita por Kent Beck, quando da publicação da segunda edição do seu livro em 2004, citado anteriormente.

Comment [A34]: Nas próximas seções

2.4.1 Valores do XP

A metodologia XP, em sua composição, tem definida por Kent Beck [2004] valores que descrevem os objetivos e definem critérios para se obter sucesso. Os valores de XP são:

- Comunicação: o fator de sucesso e insucesso de projetos de *software* é atribuído em grande parte à qualidade na comunicação. A grande maioria dos processos existentes valoriza a comunicação, porém frente à dificuldade de comunicar-se aposta na documentação extensa e complexa. XP inova e ousa ao priorizar a comunicação pessoal e oral ao invés da escrita. O contato presencial, por meio de sinais sutis e linguagem corporal, possibilitam um enriquecimento da comunicação, além permitir que dúvidas sejam discutidas e resolvidas logo que surgem. Já a documentação escrita sempre tende a desatualizar-se rapidamente.

- **Simplicidade:** a partir do Princípio de Pareto – 80% das consequências são frutos de 20% das causas –, os desenvolvedores adotam que se produza o mais simples possível que seja funcional. XP recomenda que manter o sistema simples é essencial para não gerar mais trabalho. Os desenvolvedores devem pensar no presente, não generalizar sem necessidade nem supor necessidades, bem como alertas a oportunidades de refatorar o software com o objetivo de simplificá-lo.
- **Feedback:** é valorizado, pois para poder mudar o plano e se adaptar, precisa saber rapidamente e com exatidão o que está acontecendo. Ao longo do desenvolvimento, é muito importante que exista *feedback* dentro do time de desenvolvimento e com o cliente e/ou demais envolvidos, para avaliar se as necessidades acordadas estão sendo atendidas. As entregas frequentes do software funcionando, por exemplo, possibilita que o cliente entenda efetivamente o que precisa, muda de idéia, ou descubra requisitos dos quais não estava ciente.
- **Coragem:** quando se quer iniciar o desenvolvimento de um software muitos são os medos envolvidos. O XP combate esses medos ao fornecer o suporte necessário para que os envolvidos possam sentir coragem para agir e tomar decisões. O time pode ter coragem de refatorar, pois sabe que os testes irão detectar erros imediatamente. O cliente pode decidir com mais coragem, quando avalia o software funcional após cada *release*, sabendo que pode priorizar as funcionalidades que lhe são mais importantes. No jogo do planejamento as estimativas das funcionalidades poderão ser feitas com mais coragem e confiança quando do tempo de entrega.
- **Respeito:** valorizar a relação entre membros da equipe e, também, a de cada membro com o projeto e sua instituição. Esse valor também é corroborado com o princípio da diversidade e visto através da prática do código compartilhado.

Comment [A35]: Não seria melhor traduzir?

Os valores têm sua reafirmação nos princípios, os quais descreveremos adiante.

2.4.2 Princípios do XP

Os princípios em XP compõem uma das bases (valores, princípios e práticas) de sua sustentação. Após alguns anos de *feedback* Kent Beck [2004] evidencia essa pilastra como um instrumento importante, também, na adaptação de práticas ao contexto local e até mesmo na criação de novas práticas. Os princípios agem como guias, específicos ao domínio da programação, para localizar práticas concretas em harmonia com os valores abstratos. Desse modo,

Kent Beck [2004] elenca os seguintes princípios do XP:

- **Humanidade:** reconhece que são pessoas com necessidades humanas que desenvolvem *software*. Destas necessidades, algumas podem ser satisfeitas no trabalho. São elas segurança, crescimento, identidade com o grupo, realização, intimidade e privacidade. Este princípio se concretiza na prática de trabalho energizado. Com tempo para se dedicar a satisfazer suas outras necessidades fora do trabalho, os seres humanos podem voltar com energia para se dedicar ao trabalho. Um dos desafios proposto por este princípio é balancear as necessidades individuais com as da equipe.
- **Economia:** busca garantir valor para o negócio. Ao economizar pensamos sobre o valor do dinheiro o longo do tempo e como melhor empregá-lo. É importante receber o mais cedo possível e gastar o mais tarde possível. É importante também refletir sobre o valor de opções que podemos tomar pela equipe e pelo sistema, percebendo que a habilidade de poder mudar de idéia no futuro deve guiar nossas decisões. Este princípio é evidenciado nas práticas de *design* incremental e pague pelo uso e também na priorização e estimativa de histórias, garantindo que uma equipe XP não invista em flexibilidade especulativa.

- **Benefício mútuo:** é o princípio mais importante e difícil de seguir. Um exemplo de como ele se concretiza é a ênfase dada em XP a testes, refatorações e a metáfora no lugar de extensa documentação escrita. A documentação não traz benefício para os programadores no ato de sua criação, só organização em um possível futuro. Enquanto investir em testes, refatorações e na construção de uma metáfora trazem benefícios imediatos aos programadores, ao sistema, ao cliente e à organização.
- **Auto-semelhança:** diz que se deve copiar estruturas e processos existentes para resolver problemas em diferentes contextos ou escalas. É o caso do ritmo similar que se observa nas práticas do ciclo de estação, ciclo semanal e nas atividades diárias de programação. Observamos que primeiro criamos testes e depois trabalhamos para que eles funcionem. No ciclo mensal, escrevemos testes de aceitação, que no final devem todos passar para validarmos um *release*. Nas atividades diárias escrevemos testes de unidade para validar tarefas que devem ser codificadas.
- **Melhoria:** indica que não devemos esperar a perfeição, mas sim fazer o melhor que podemos hoje, para poder fazer o melhor amanhã. A prática de ciclo de estação evidencia este princípio dando à equipe a oportunidade de melhorar o plano de um *release*. A prática de *design* incremental também segue o princípio da melhoria.
- **Diversidade:** lembra que um time com pessoas diferentes apresenta mais habilidades, conhecimentos e oportunidades. A diversidade é causa de conflitos, sendo importante lidar com essa possibilidade valorizando o respeito. O princípio é evidente nas práticas do time completo e nos diferentes ciclos de planejamento. Pessoas com perspectivas diversas têm igual oportunidade de colaborar nestes: aquelas que pensam em longo prazo contribuindo com o ciclo de estação e as que têm perspectivas de curto prazo contribuindo com o ciclo semanal.
- **Reflexão:** implica em pensar sobre como e por que trabalhamos. Este princípio pode guiar uma equipe a adotar práticas como a retrospectiva e a realizar análises frequentes do seu processo de adoção da metodologia. Para isso, é preciso tempo para pensar. É importante socializar com a equipe em contextos de diversão ou até em refeições. A reflexão é evidenciada nas práticas do ciclo de estação e o ciclo semanal, nas conversas de pares programando e na prática de integração contínua.
- **Fluxo:** determina que exista uma corrente contínua de atividades e que o processo deve explicitá-la. Desta maneira permiti-se que as etapas do desenvolvimento aconteçam em paralelo e não sequencialmente como proposto em metodologias mais tradicionais. As práticas de *releases* frequentes e integração contínua evidenciam isto.
- **Oportunidade:** nos leva a encarar problemas como oportunidades para mudança. Estar aberto a oportunidades de aprender e melhorar durante todo o processo é importante.
- **Redundância:** aumenta as nossas chances de sucesso, promovendo várias oportunidades de fazer a coisa certa. A redundância está presente na complementação das práticas, e nos testes de unidade automatizados: escrevemos o código fonte e, de maneira redundante, escrevemos mais código para verificar se o primeiro funciona, diminuindo a nossa probabilidade de errar.
- **Falha:** indica que pode ser bom falhar, desde que se aprenda com a experiência. Quando uma equipe não sabe para onde ir, arriscar-se a falhar pode ser o caminho mais curto para obter o sucesso. Este princípio é complementar ao valor de coragem e se evidencia na prática de abandonar código e começar de novo quando percebemos que determinado plano não poderá ser realizado.

- **Qualidade:** sempre presente e em alta. Este princípio diz que quanto maior a qualidade, mais fácil será realizar o trabalho. Ele complementa o princípio da humanidade ao satisfazer a necessidade de se orgulhar do trabalho feito e é evidenciado na prática de controle do escopo no planejamento.
- **Passos pequenos:** garante que iremos fazer sempre o caminho mais curto na direção correta, pois a execução de tarefas complexas em passos pequenos diminui o risco. A prática de *design* incremental, integração contínua, implantação incremental e implantação diária refletem este princípio.
- **Aceitação de Responsabilidade:** evidencia que só o próprio indivíduo pode se responsabilizar por suas ações. Este princípio está claro na prática de estimativas feitas pelos próprios programadores no jogo do planejamento.

Abordaremos as práticas, que dão maior visibilidade e aplicabilidade aos valores e princípios, na seção seguinte.

2.4.3 Práticas do XP

Extreme Programming define também práticas que tornam o processo viável e possível de seguir seus valores e princípios. As práticas são simples, porém o poder da metodologia provém da combinação delas. As práticas abordadas a seguir representam uma evolução em relação à primeira versão do XP. Nessa versão, Kent Beck [2004] as categoriza em dois tipos – primária e corolário, a saber:

As práticas primárias são guias para se começar a adoção de programação extrema em uma organização. Elas podem ser implantadas facilmente, pois são seguras e devem ser introduzidas em pequenos passos, para evitar uma mudança muito rápida na cultura da organização. É importante ter tempo para que os novos hábitos sejam incorporados pela equipe. A seguir será detalhado as novas práticas.

- **Sentar Juntos:** deixa explícita a necessidade de se ter um espaço onde toda a equipe possa trabalhar junta, valorizando a comunicação e possibilitando que as pessoas possam se beneficiar de todos seus sentidos ao conversar. Ressalta-se também a necessidade de pequenos espaços privativos para respeitar o princípio de humanidade.
- **Time Completo:** lembra que a equipe precisa de pessoas com todas as habilidades e perspectivas necessárias para o sucesso do projeto. Nesta segunda versão, Beck remove a limitação de tamanho das equipes e ressalta somente dois limites naturais: equipes de 12 pessoas é o limite natural de pessoas com as quais um indivíduo pode interagir em um dia; e equipes de 150 pessoas é o limite natural para que um indivíduo se lembre de todos outros no time. Beck recomenda que se fracione problemas complexos e que estes sejam resolvidos com times que respeitem estes limites.
- **Espaço de trabalho informativo:** era uma prática implícita que ganhou destaque na segunda edição. Ela diz que qualquer observador interessado deve ser capaz de olhar para o espaço de trabalho e ter idéia do andamento do projeto em pouco tempo. O espaço informativo deve conter cartazes grandes e visíveis, que comunicam medidas coletadas pelo acompanhador. Limpeza, ordem, espaço para programação pareada e disponibilidade de água e comida garantem que o espaço informativo complemente a prática de trabalho energizado.
- **Histórias que antes faziam parte do jogo do planejamento,** viram uma prática independente na segunda versão. Elas continuam iguais: textos simples que expressam necessidades do cliente e que são estimadas, o quanto antes, pelos programadores.
- **Ciclo semanal** que antes era parte do jogo do planejamento, explicita as iterações que fazem parte de um *release*. O ciclo semanal inclui um jogo de planejamento do trabalho que deve ser efetuado em

aproximadamente uma semana, ou seja, planeja as iterações de menor granularidade. Uma reunião no começo da semana revê o progresso de um *release*, comparando o que foi feito com o que tinha sido planejado na semana anterior. O cliente prioriza uma semana de histórias, que são divididas em tarefas, que por sua vez são aceitas e estimadas pelos programadores. A semana começa com a escrita de testes de aceitação da iteração e acaba com a implantação do sistema codificado. Beck reconhece que o planejamento custa tempo, mas que esse é um gasto necessário.

- **Ciclo de estação:** que antes era parte do jogo do planejamento, explicita o planejamento de um *release*. A avaliação de um ciclo de estação pode identificar gargalos e dependências da equipe com outras partes da organização, apresentando uma oportunidade para melhorias e reparos. Durante o planejamento deste ciclo, temas são escolhidos para a estação. Estes temas servem para agrupar histórias relacionadas, que também são criadas neste momento. O uso de temas garante que não há exagero de detalhes e que o planejamento acontece com a perspectiva do longo prazo. O enfoque do planejamento da estação é também perceber como o projeto se encaixa com o resto da organização. A estação pode ter duração variável e depende do contexto do negócio. Em muitas organizações um trimestre é uma boa medida para avaliar o progresso.
- **Folga:** reconhece que o planejamento, por melhor que tenha sido, sempre falha. Para evitar atrasos ou a necessidade de renegociação de escopo, o planejamento deve conter explicitamente espaços de folga. Isto pode acontecer tanto com a introdução de tarefas menores, que podem ser descartadas em caso de atraso, quanto com a distribuição de histórias de trabalho livre para os programadores. Se o progresso de um ciclo for bom, este tempo pode ser usado para pesquisa e para manter o trabalho energizado. Caso atrasos ocorram, a folga pode ser descartada e mais trabalho pode ser realizado para que a equipe consiga entregar as histórias que prometeu ao cliente.
- **Build veloz:** exige que o sistema deve ser compilado por completo e todos os testes devem ser executados, de maneira automática, em no máximo 10 minutos. Esta prática provê agilidade equipe e complementa a habilidade de entregar *releases* pequenos. O limite de 10 minutos é somente o tempo razoável para que o par possa tomar um café durante o build, porém não é essencial. Se o build demora menos que 10 minutos, excelente, o café pode ficar para depois. Se demora mais existem duas possibilidades. A primeira é de que o build pode ser refatorado, para que não rode todos os testes de aceitação por exemplo, pois estes podem demorar muito tempo para serem executados. A segunda é de que o build é lento pois o sistema é muito complexo, e a demora pode ser um indício da necessidade de simplificá-lo.
- **Design incremental:** a prática de *design* simples rendeu críticas a XP que diziam que a metodologia não investia em *design* da aplicação. Para responder às críticas, Beck a redefine como *design* incremental. Esta prática implica em investimento diário no *design* da aplicação justificando seu fluxo contínuo ao mostrar que, se realizado próximo de ser utilizado, ele tende a ser mais eficiente e valioso.

Dente as práticas do XP duas não sofreram modificações e compõem as primárias. A programação pareada na qual todo o código é produzido por pares de programadores, cada par trabalhando na mesma máquina. E a integração contínua cujo o código é integrado, o *build do software* é gerado e os testes são executados várias vezes ao dia.

As práticas corolário são difíceis ou perigosas de serem implementadas sem antes serem dominadas as práticas primárias. O conselho de progredir incrementalmente em direção às práticas se mantém. A prática do cliente sempre presente foi renomeada para envolvimento real com o cliente e assume que muitas vezes um cliente direto não poderá ser colocado com a equipe. A prática de propriedade coletiva do código foi renomeada para código compartilhado e se mantém como prática corolário. A seguir iremos detalhar as novas práticas.

- **Implantação incremental:** lida com uma equipe que deve substituir um sistema legado e diz que essa substituição deve ser feita incrementalmente. O importante é manter o sistema funcionando e ter segurança

na migração. É possível que, durante algum tempo, ambos o legado e o novo sistema devam funcionar em conjunto, adicionando um pouco de trabalho de comunicação extra tanto no sistema quanto com usuários, mas garantindo harmonia na migração.

- Continuidade da equipe: propõe que equipes eficientes continuem trabalhando juntas. Deve-se incentivar um rodízio razoável entre equipes, mas ao se concentrar na eficiência da organização como um todo, o valor de equipes que trabalhem bem juntas se torna evidente.
- Redução da equipe: com o passar do tempo, a necessidade de uma equipe grande pode diminuir, principalmente se o sistema entra em um ciclo de manutenção. Se isto acontecer, mantenha a carga de trabalho constante e distribua tarefas de modo a deixar alguém ocioso; esta pessoa pode ser liberada para formar novos times. Esta prática tende a eliminar o desperdício e ajudar a organização a resolver novos problemas.
- Análise da causa inicial: sempre que encontrar um defeito, elimine o defeito e sua causa, para que o mesmo tipo de erro não ocorra novamente. Esta prática define uma nova atividade: ao encontrar um defeito, um par deve primeiro escrever um teste de aceitação automatizado que evidencie o erro no nível do sistema e então escrever um teste de unidade no menor escopo possível. O par deve proceder para resolver o problema e passar nos testes. O último passo é descobrir por que o defeito surgiu e, principalmente, como ele passou despercebido. A técnica de análise da causa inicial propõe que se pergunte 5 vezes o motivo do defeito ter surgido e sugere que a causa inicial, na maioria das vezes, é um problema relacionado à equipe e que pode ser resolvido utilizando-se práticas que evitem que ele recorra.
- Código e testes: explicita que só o código fonte e os testes automatizados devem ser artefatos permanentes gerados por uma equipe XP. Até mesmo as histórias e cartazes devem ser descartados, pois o histórico do projeto se mantém por mecanismos sociais. A cerimônia envolvida em documentação interrompe o fluxo de valor.
- Repositório único de código: complementa a prática de código compartilhado e vai além, dizendo que todo o código deve estar contido em um único repositório que não deve ter branches permanentes. O problema com múltiplos repositórios é que eles não escalam. Se o seu sistema é tão complexo que precisa de repositórios separados, isso é evidência de um problema no *design*.
- Implantação diária: é a evolução da prática de *releases* pequenos e é ainda mais extrema. Ela determina que o seu sistema deve ser implantado diariamente, de preferência com auxílio do build veloz. O objetivo é colocar histórias implementadas em produção toda noite, para que os usuários possam usufruir benefícios o quanto antes. Esta prática depende de uma baixa taxa de defeitos e de um processo automático de implantação, com habilidade para implantação incremental e eventual *rollback* em caso de falhas.
- Contrato de escopo negociável: determina como devem ser feitos contratos em um projeto que adota XP, fixando o tempo, custos e a qualidade, mas mantendo o escopo negociável. Desta maneira, o escopo é negociado constantemente com o cliente, possivelmente no planejamento do ciclo de estação. Assim a equipe poderá celebrar uma sequência de contratos curtos.
- Pague pelo uso: também aborda o lado de negócios de um projeto XP, sugerindo que o cliente pague por toda vez que for usar o sistema (trazendo à tona o debate sobre “arquiteturas orientadas a serviços” que está em voga na indústria atualmente). Esta prática valoriza o dinheiro como *feedback* mais importante e provê ao cliente possibilidades de prever custos.

O quadro abaixo ilustra e converge o entendimento das práticas da primeira versão do XP com a segunda.

Quadro 1 - Práticas do XP 1ª versão vs 2ª versão

Práticas do Extreme Programming		
1ª VERSÃO 1999	2ª VERSÃO 2004	Categoria 2ªv
O jogo do planejamento	Histórias Ciclo semanal Ciclo de estação	Primária Primária Primária
Releases pequenos	Implementação incremental Implementação diária	Corolário Corolário
Metáforas Projeto de <i>software</i> simples Refatoração	<i>Design</i> incremental	Primária
Teste	Desenvolvimento orientado por testes Código e testes	Primária Corolário
Programação em pares	Programação Pareada	Primária
Propriedade coletiva Padrão de codificação	Código compartilhado Repositório único de código	Corolário Corolário
Integração contínua	Integração contínua	Primária
-	Área de trabalho informativa	Primária
-	Análise de causa inicial	Corolário
40 horas semanais	Folga Trabalho energizado	Primária Primária
Cliente no local	Sentar junto Time completo Envolvimento real com o cliente	Primária Primária Corolário
-	Continuidade do time	Corolário
-	Redução do time	Corolário
-	Build de 10 minutos	Primária
-	Contrato de escopo variável	Corolário
-	Pague pelo uso	Corolário

2.4.4 Papéis do XP

Nesta nova versão, Beck inclui todas as funções que tipicamente se encontram em uma organização que desenvolve *software* e explicita quais são os papéis que cada uma pode assumir para colaborar com uma equipe XP. A diversidade de papéis traz benefício à prática de time completo. Cada parte no grupo deve entender seu papel no todo e a interação entre as pessoas deve seguir os princípios de fluxo e benefício mútuo. Uma pessoa pode assumir mais de um papel e os papéis podem ser revezados entre as pessoas.

Vimos que Extreme Programming sugere que o time seja multidisciplinar com habilidades necessárias para realizar o projeto. A primeira versão de XP era mais voltada aos programadores enquanto na segunda versão é dado maior valor a todos os outros papéis dentro da equipe. Os principais papéis em XP são [BECK 2004]:

- Programadores: segundo Beck [2004] este é o coração de XP. Responsável por quebrar histórias e tarefas, escrever testes e código e automatizar processos manuais. Existem dois papéis especiais para programadores, aqueles com mais experiência atuam com *Coach* ou líderes de equipe que auxiliam os menos experientes da equipe, enquanto o *Tracker* atua coletando e compartilhando dados sobre o projeto e do processo.

- **Arquitetos:** executam refatoração de larga escala, escrevem testes de carga automatizados para definir cenários de estresse e auxiliam os programadores no particionamento do sistema mantendo a ênfase no projeto de alto-nível.
- **Analista de Testes:** trabalham junto com clientes e analistas de negócios para escrever testes de aceitação automatizados definindo cenários de sucesso e falha em cada história. Estes treinam os programadores a utilizarem as ferramentas de teste.
- **Analista de Negócios:** trabalham com clientes para definir as histórias do sistema e auxiliam os programadores a definir o valor da importância das mesmas.
- **Projetistas de Interação:** avaliam o modo como o sistema está sendo utilizado pelos usuários finais, assim podem ser levantadas novas histórias bem como propostas de melhorias na interface gráfica.
- **Gerente de Projetos:** facilitam a comunicação dentro da equipe, removendo impedimentos e coordenando a comunicação com as pessoas externas a equipe.
- **Gerente do Produto:** escrevem e priorizam histórias do ciclo semanal e fecham o tema para o ciclo trimestral. Encorajam a comunicação entre a equipe de desenvolvimento e o cliente para que suas necessidades mais urgentes sejam atendidas de imediato.
- **Usuários:** Por utilizar o sistema diariamente, ajudam o time a escrever e escolher de forma melhor as histórias do sistema. Por fornecerem as necessidades do sistema é ideal que tenham experiência com sistemas similares.

É importante notar que os papéis não são fixos e que cada um deve contribuir com tudo que pode para a equipe.

2.4.5 Ciclo de vida do projeto XP

Um software desenvolvido a partir do XP terá que percorrer algumas fases durante o seu ciclo de vida. De acordo com o tipo de projeto ou a característica da organização, por exemplo, essas fases podem sofrer modificações, porém manterá forte semelhança estrutural. Em cada fase várias atividades são realizadas. Um projeto XP passa basicamente pelas seguintes fases: exploração, planejamento, iterações e aprovação. Abaixo descreveremos algumas das principais fases de um projeto e conseqüente visão de como ele acontece.

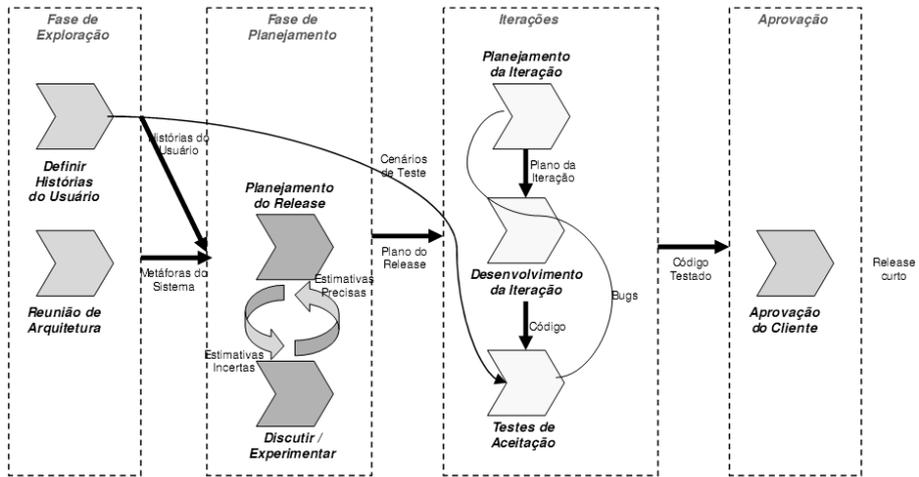
Na fase de exploração, anterior à fase de construção do sistema, as investigações são feitas e é verificada a viabilidade de possíveis soluções a serem implementadas. Os programadores elaboram possíveis arquiteturas e tentam visualizar como o sistema funcionará considerando os mais diversos aspectos. Ao tempo em que o cliente prepara as histórias. Conseqüente, os programadores e os clientes vão ganhando confiança, e quando eles possuem histórias suficientes, passam a formatar o primeiro *release* do sistema.

A fase de planejamento atende o momento em que será acordada uma data lançamento do primeiro *release*. Nessa etapa os programadores de posse das histórias elaboradas pelo cliente assinalam certa dificuldade para cada uma e, baseados na sua velocidade de implementação, dizem quantas podem implementar em uma iteração, denominado de planejamento da iteração. Depois, os clientes escolhem as histórias de maior valor de

negócio para serem implementadas na iteração. O processo então se repete até terminar as iterações do *release*. O tempo para cada iteração deve ser de uma a três semanas e para cada *release* de dois a quatro meses. Na fase das iterações do *release*, de posse do planejamento da iteração, o plano de iteração é posto em desenvolvimento, momento em que os programadores seguem um fluxo de atividades (casos de testes funcionais/unidade, projeto e refatoramento, codificação, realização dos testes e integração entre outras) e, ainda, submetido aos testes de aceitação. Os testes de aceitação já foram escritos a partir das histórias do cliente. À medida que esse fluxo vai sendo seguido, o sistema vai sendo construído segundo os princípios, valores e práticas apresentados nas seções anteriores.

Na fase de aprovação o cliente recebe algumas das histórias acordadas para o *release* já em funcionamento. Nesse momento o cliente analisa o produto entregue e aprova ou desaprova. Qualquer que seja o posicionamento do cliente, essas informações serão muito úteis às demais *releases* e ao referido ciclo.

A figura X a seguir corrobora com o entendimento do ciclo de vida do produto XP.



Comment [A36]: Que figura??

Comment [A37]: E a fonte??

Em algumas abordagens são citadas mais duas fases: manutenção e morte.

A fase de manutenção pode ser vista como uma característica intrínseca a um projeto XP. O fluxo iterativo e incremental caracteriza o produto em constante manutenção, a medida que novas funcionalidades surgem, tecnologias e pessoas passam a incorporadas e o melhorando o código.

A fase de morte corresponde ao término de um projeto XP. Um projeto XP chega ao seu fim de duas maneiras: a primeira é atendendo as necessidades do cliente e com um produto de qualidade, e a segunda por motivos negativos de inviabilidade econômica, dificuldade de adicionar funcionalidades a baixo custo e/ou alta presença de erros entre outras.

2.6. Scrum

A metodologia ágil *Scrum* foi criada em 1996 por Ken Schwaber e Jeff Sutherland e destaca-se das demais metodologias ágeis pela maior ênfase dada ao gerenciamento do projeto. Reúne atividades de monitoramento e *feedback*, em geral, reuniões rápidas e diárias com toda a equipe, visando à identificação e correção de quaisquer deficiências e/ou impedimentos no processo de desenvolvimento. [SCHWABER 2008]

Scrum vem sendo largamente utilizado em organizações ao redor do mundo. Ele permite manter o foco na entrega do maior valor de negócio, no menor tempo possível permitindo à rápida e contínua inspeção do software em produção. As necessidades do negócio é que determinam as prioridades do desenvolvimento de um sistema. As equipes se auto-organizam para definir a melhor maneira de entregar as funcionalidades de maior prioridade. Portanto, entre cada duas a quatro semanas todos podem ver o software real em produção, decidindo se o mesmo deve ser liberado ou continuar a ser aprimorado.

2.6.1 Características do Scrum

De acordo com [SCHWABER e BEEDLE, 2002], *Scrum* trata-se de uma abordagem empírica de lidar com o caos, em detrimento a um processo bem definido. Focado em pessoas para ambientes em que há requisitos voláteis, resultando em uma abordagem que reintroduz as idéias de flexibilidade, adaptabilidade e produtividade. O foco da metodologia é encontrar uma forma de trabalho dos membros da equipe para produzir o software de forma flexível e em um ambiente em constante mudança.

A metodologia baseia-se em princípios como: equipes pequenas, no máximo sete pessoas, requisitos instáveis ou desconhecidos e iterações curtas. Cada ciclo do *Scrum* é denominado *Sprint*, que possui intervalos de tempo reduzido de 15 a 30 dias. Esta metodologia não requer ou fornece qualquer técnica ou método específico para o desenvolvimento do software, ela enfatiza o planejamento e gerenciamento dos projetos, através de um conjunto de regras e práticas gerenciais que são estabelecidas.

No *Scrum* o cliente torna parte da equipe de desenvolvimento, através da figura do **PO**, e existem reuniões freqüentes com todos os envolvidos no projeto.

Comment [A38]: O que é PO??

3.6.2 Papéis do Scrum

O Scrum define três papéis principais para as diferentes tarefas, propósitos do processo e suas práticas: *Scrum Master*, *Product Owner* (PO) e Time. [SCHWABER, 2008].

• **Scrum Master:** O *Scrum master* gerencia o processo, dissimulando o *Scrum* a todos os envolvidos no projeto e adequando a metodologia à cultura da organização. Seu papel é remover os impedimentos do projeto e garantir que todos do time sigam as regras e práticas do Scrum.

Ele é o líder e facilitador para o time e *Product Owner*, responsável por: resolver barreiras entre o time e o PO; ensinar o cliente a aumentar o retorno sobre o investimento; garantir que o processo seja seguido; motivar e incentivar a equipe de desenvolvimento, facilitando a criatividade e a capacitação; melhorar a produtividade da equipe; melhorar as práticas de engenharia e prover ferramentas de modo que cada nova funcionalidade seja **potencialmente realizada; manter e divulgar intra-time as informações sobre os progressos da equipe.**

Comment [A39]: Ele quem??

• **Product Owner:** O *Product Owner* (PO) representa o cliente no projeto. Seu foco é na parte comercial do produto. Ele é o representante de todos os *stakeholders*. O *PO* define os objetivos do projeto criando requisitos iniciais e gerais (*Product Backlog*), planeja as entregas e prioriza o *Product Backlog* a cada *Sprint*, garantindo que as funcionalidades mais importantes sejam construídas prioritariamente. [SZALVAY, 2007].

Desafios de um *Product Owner*:

1. Não gerenciar a equipe. Isto é especialmente desafiador se alguns membros da equipe requisitam sua intervenção para questões que devem se resolver por si;
2. Não acrescentar mais funcionalidades após a *Sprint* já estar em andamento, somente em casos devidamente justificados;
3. Equilibrar os interesses dos *Stakeholders*.

Segundo [SCHWABER, 2009], o *Product Owner* pode ser alguém do Time, trabalhando também em desenvolvimento. Mas essa missão adicional pode reduzir a sua habilidade de lidar com as partes interessadas. No entanto, o *Product Owner* nunca pode ser o *ScrumMaster*.

• **Time:** O time é um grupo de pessoas, com diferentes habilidades, necessárias para implementar as funcionalidades, envolve analistas, desenvolvedores, designers, gerente de qualidade, entre outros. Quando necessário, a equipe tem a autoridade de decidir as ações que serão realizadas e priorizá-las organizando-as nas Sprints. O time deve gerenciar seu próprio trabalho, sendo responsáveis coletivamente pelo sucesso do projeto. De acordo com [SCHWABER, 2009] as pessoas que se recusam a programar porque são arquitetas ou designers não se adaptam bem a Times. Todos contribuem, mesmo que isso exija aprender novas habilidades ou lembrar-se de antigas. Não há títulos em Times, e não há exceções a essa regra. Os Times também não contém sub-times dedicados a áreas particulares como testes ou análise de negócios.

2.6.3 Práticas do Scrum

A seguir, serão apresentadas as práticas do *Scrum* de acordo com Schwaber e Beedle (2002);

Comment [A40]: Não seria [2002] ??

- **Product Backlog:** define tudo o que é necessário no produto final baseado no conhecimento atual. É o ponto de partida do *Scrum* onde são definidas as funcionalidades, as prioridades, a tecnologia e as estratégias. A definição destes itens pode ser feita por qualquer pessoa ou setor envolvidos no projeto. Para algum item ser adicionado ao *Product Backlog* é necessário que a equipe esteja de acordo. Os itens do *Product Backlog* são documentados com as seguintes informações: uma descrição sucinta, uma estimativa, que deverá sempre ser estipulada em horas, um responsável e uma prioridade (muito alta, alta e média). Para facilitar a visualização é sugerido que os itens sejam separados por prioridade.
- **Sprint:** Considerada a parte mais importante do *Scrum*. É onde são executados os itens definidos no *Product Backlog*. Cada *Sprint* deve durar no máximo 30 dias. Não existem processos pré-definidos dentro de uma *Sprint*, mas sim reuniões periódicas (Reuniões Diárias do *Scrum*) que coordenam a realização das atividades existentes. É criado um *Sprint Backlog* com os itens selecionados para serem trabalhados na *Sprint* corrente. Ao final de cada *Sprint*, segundo Beedle (1998), é criada uma pequena versão do software, com o objetivo de mostrar ao cliente o que está sendo desenvolvido. Na *Sprint* acontece a integração da parte software que foi desenvolvida com as outras partes já implementadas e são feitos testes, garantindo um progresso real das atividades. Uma nova *Sprint* começa com um novo conjunto de itens no *Sprint Backlog*, sendo assim, dentro do desenvolvimento *Scrum*, podem acontecer N Sprints. Schwaber (1995) ressalta que o desenvolvimento dos itens dentro de uma *Sprint* não segue nenhum processo pré-definido, enquanto Abrahamsson (2002) sugere o uso das fases tradicionais de desenvolvimento de software: análise, projeto, implementação, testes e entrega. Outra sugestão seria a combinação de outros métodos ágeis que possuem processos definidos nesta fase.
- **Sprint Backlog:** É o ponto de partida para cada *Sprint*. Consiste em um conjunto de funcionalidades selecionadas do *Product Backlog* para serem implementadas durante a *Sprint*. Os itens são selecionados pelo *Scrum Team* em conjunto com o *Scrum Master* e o *Dono do Produto* na Reunião de Planejamento da *Sprint*, de acordo com as prioridades dos itens. Quando todos os itens do *Sprint Backlog* estiverem prontos, um novo incremento do sistema é entregue.
- **Reunião de Planejamento da Sprint:** Cada *Sprint* inicia com uma reunião chamada de Reunião de Planejamento da *Sprint*, que tem por objetivo analisar os itens do *Product Backlog* a fim de priorizá-los para o desenvolvimento e, assim, definir o *Sprint Backlog*.
 - **Reuniões Diárias do *Scrum*:** De acordo com Linda (2000), as Reuniões Diárias do *Scrum* são realizadas diariamente ou em dias alternados e possuem duração de 15 a 30 minutos, no máximo. Este tempo é suficiente para identificar os problemas, mas não para definir soluções. Discussões para resolução de obstáculos são feitas mais tarde, onde somente os envolvidos no problema participam. Segundo Schwaber e Beedle (2002), durante as Reuniões Diárias do *Scrum* são levantadas as três seguintes questões para cada membro da equipe:
 - O que foi finalizado desde a última reunião? (São registradas quais tarefas foram completadas e quais ainda estão pendentes).

- Quais as dificuldades encontradas durante o trabalho? (São registradas todas as dificuldades encontradas para mais tarde encontrar uma maneira de resolvê-las).
- Quais atividades pretende-se realizar até a próxima reunião? (São escolhidas as tarefas mais importantes. Devido ao curto espaço de tempo entre as reuniões, as tarefas são geralmente pequenas). A reunião também possibilita que todas as pessoas fiquem informadas sobre o progresso e as dificuldades encontradas.

• **Revisão da *Sprint*:** No último dia de cada *Sprint*, o *Scrum Team* e o *Scrum Master* apresentam o incremento para o cliente, gerente e Dono do Produto numa reunião informal. Os participantes avaliam o incremento do produto e decidem sobre as atividades seguintes. Na reunião poderão ser adicionados novos itens ao *Product Backlog*.

2.6.4 Ciclo de Vida do Scrum

O *Scrum* possui um ciclo de vida composto por três fases:

- **Planejamento (*Pre-game phase*):** Os requisitos são produzidos em um artefato denominado *Product Backlog*. É estabelecida a visão do projeto e expectativas garantindo recursos para a sua execução, como equipe de desenvolvimento e ferramentas.
- **Desenvolvimento (*game phase*):** Nesta fase o sistema é desenvolvido em *Sprints* em que novas funcionalidades são adicionadas. Cada um desses ciclos é desenvolvido de forma tradicional, ou seja, primeiramente faz-se à análise, em seguida o projeto, implementação e testes. Todas as *Sprints* têm como resultado um incremento do produto final que é **potencialmente entregável** e cada uma começa imediatamente após a anterior.
- **Releasing (*post-game phase*):** Nesta fase é realizado a entrega do produto ao cliente.

Comment [A41]: Acho que não tá legal o termo utilizado

2.6 FEATURE DRIVEN DEVELOPMENT

O *Feature Driven Development* (FDD) é um processo ágil para gerenciamento e desenvolvimento de software, criado em 1977 em um grande projeto em Java para a *Unided Overseas Bank*, em Singapura. Nasceu a partir da experiência de análise e modelagem orientadas por objetos de Peter Coad e de gerenciamento de projetos de Jeff De Luca, frente a uma necessidade da referida instituição [SLIGER 2008].

A FDD é uma metodologia voltada para o cliente e orientada a modelagem, combinando algumas das melhores práticas do gerenciamento ágil de projetos com uma abordagem completa para Engenharia de Software orientada por objetos [COAD 1999]. Compõe de um arcabouço particular, através de seus princípios e práticas, que proporciona um equilíbrio entre as filosofias tradicionais e as ágeis. O referido equilíbrio, segundo Michele Sliger [2008] proporciona uma transição mais suave para organizações mais conservadoras, e a retomada da responsabilidade para as organizações que se desiludiram com as propostas mais radicais. Stephen Palmer [2002] coloca a FDD como sendo algo que será incorporado a sua empresa com fácil adaptação e que possibilitará resultados frequentes, tangíveis e funcionais. Na seção seguinte iremos abordar pontos que facilitarão na escolha da metodologia a ser aplicada em determinado projeto.

2.6.1 Características do FDD

As metodologias ágeis devem seguir o todo ou parte do que foi acordado no Manifesto Ágil, mesmo surgidas antes do referido manifesto, e por isso tendem a possuir características comuns [BECK, FOWLER 2001]. Algumas características intrínsecas nos processos ágeis são levantadas por Pekka Abrahamsson [2003], a saber:

- Cliente presente;

- Iterações curtas;
- Equipes pequenas (menos de 12 aproximadamente);
- Entregas frequentes do produto;
- Adaptativos as mudanças;
- Flexibilidade;
- Simplicidade.

Porém, Craig Larman [2003] destaca que dentre as características comuns aos processos ágeis sempre existirá particularidades que as diferenciem.

Seguindo essa linha das características ágeis, temos alguns pontos que particularizam o FDD [PALMER 2002]:

- Resultados úteis a cada duas semanas ou menos;
- Blocos bem pequenos de funcionalidade valorizada pelo cliente, chamados "*features*";
- Planejamento detalhado e guia para medição;
- Rastreabilidade e relatórios com incrível precisão;
- Monitoramento detalhado dentro do projeto, com resumos de alto nível para clientes e gerentes, tudo em termos de negócio;
- Fornece uma forma de saber, dentro dos primeiros 10% de um projeto, se o plano e a estimativa são sólidos.

Essas características poderão ser observadas e, também, outras novas identificadas ao longo das seções que abordam sobre o FDD.

2.6.2 Papéis do FDD

O FDD apresenta em seu escopo a definição de papéis para que se possa ter uma maior organização e visão na hora de iniciar um projeto. Nesse contexto, o FDD estrutura seu time [PALMER 2002] em:

- Gestor do Projeto: trata das questões financeiras e administrativas do projeto. É o membro que decide sobre o escopo, objetivos, o time e prazos, no que se refere à decisão final. É, também, atribuição sua prezar por ótimas condições de trabalho e manter o time focado, com vistas a maximizar os resultados;
- Chefe de Design: responsável por toda a arquitetura do projeto, bem como das sessões de design, nas quais apresenta seus entendimentos ao time;
- Gestor de Desenvolvimento: acompanha as atividades de desenvolvimento do código diariamente, bem como a incumbência de fazer com que problemas não cheguem ao time ou que o mesmo seja resolvido o mais rapidamente. Desempenha suas funções afinado com o gestor de projeto;
- Programador Chefe: é responsável por uma equipe pequena no que se refere a divisão e atribuição de trabalho entre seus membros. Recomenda-se que seja um programador experiente, pois fará parte de suas atribuições a escolha das funcionalidades a serem implementadas em cada iteração, bem como o relatório de atividades do time. Deve permitir um canal aberto de comunicação com o chefe de design e com o programador chefe;

- **Dono de Classe:** responsável pela arquitetura, implementação, teste e documentação de uma determinada classe e fará parte das equipes cujas funcionalidades sejam envolvidas a sua classe;
- **Especialista da Área:** membro conhecedor do assunto sobre o qual a aplicação atuará. Trabalha em conjunto com o gestor de projeto em algumas questões macro que sua área lhe habilita, bem como ao lado dos desenvolvedores com suporte de conhecimento necessários a construção da fature.

Por se tratar de uma metodologia ágil, na qual a flexibilidade e adaptabilidade são presentes em sua essência, um membro pode assumir mais de um papel, simultaneamente, e um mesmo papel pode ser assumido por vários membros. Isso acontece a partir das características de cada projeto.

Como a proposta do FDD foi utilizada inicialmente em um time com aproximadamente 50 pessoas, pode fazer necessário o surgimento de outros papéis para compor o time dentro da característica do projeto proposto.

O FDD, inicialmente, recomenda uma composição de equipe de até 20 membros, mas existem casos conhecidos na literatura e na prática de indústrias de software, o processo atendendo a times bem maiores.

Alguns métodos ágeis, inclusive o FDD, afirmam se aplicar a qualquer projeto de desenvolvimento ágil, sem importar suas características [ABRAHAMSONN, 2003].

Na seção seguinte será abordado a respeito das práticas recomendadas pelo FDD.

2.6.3 Práticas do FDD

O FDD possui em seu arcabouço um conjunto de boas práticas baseadas nas que identificamos e/ou vivemos na engenharia de software. As práticas do FDD focam em atender as necessidades do cliente e a produção do sistema com qualidade. Abaixo serão descritas algumas dessas práticas [PALMER 2002]:

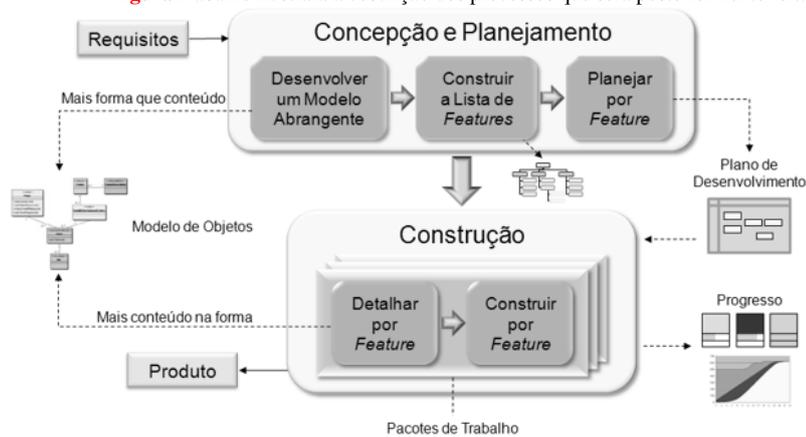
- **Modelagem de objeto do domínio:** é construída, inicialmente, uma modelagem genérica com suas funcionalidades, dentro da perspectiva da orientação a objetos. Essa modelagem possibilita um maior entendimento/visibilidade do problema a ser resolvido;
- **Desenvolvimento por funcionalidade:** as atividades a serem desenvolvidas devem ser analisadas com a perspectiva de verificar a possibilidade de serem recompostas em atividades menores. Essa prática possibilita mais segurança, maior flexibilidade e escalabilidade ao código;
- **Posse individual do código:** uma funcionalidade ou um conjunto delas é delegado a determinado desenvolvedor e este se torna automaticamente responsável por tudo que estiver relacionado ao código, desde a performance, passando pela consistência e correteude até a integração da classe;
- **Equipes de funcionalidades:** as equipes são montadas para atender determinada funcionalidade ou um pequeno conjunto delas, conforme o tamanho, dependência e semelhança. A partir do ponto de visão/entendimento de cada membro é convergido para determinar o design da funcionalidade e sua solução final;
- **Inspeções:** inspeção de código é uma prática da engenharia de software que possibilita um melhoramento do código no que se refere à redução de erros, melhor modelagem e fatores como legibilidade e alta coesão;
- **Gerência de configuração:** atividade que busca manter o controle sobre o código fonte permite vinculação referencial (funcionalidade<>código<>proprietário), além de manter histórico de alterações no código;
- **Build constante:** deve existir sempre uma versão do sistema rodando numa máquina, garantindo a equipe o funcionamento de pelo menos uma versão do sistema. Essa prática garante que existirá uma versão do sistema que pode ser utilizada, a parte, a qualquer momento sem interferir no desenvolvimento; e

- Visibilidade do progresso: a prática recomenda que um relatório de progresso das atividades do projeto seja mantido visível à equipe e aos demais interessados, para saberem exatamente como estão em termos de produtividade. Essa atividade requer muita atenção, precisão e constância, pois dados incorretos podem levar a decisões desastrosas ao projeto e demais entidades envolvidas.

A seção seguinte detalha o ciclo do sistema a luz do FDD.

2.6.7 Ciclo de Vida do FDD

O FDD possui uma estrutura muito objetiva. A sua composição apresenta-se em duas fases (Concepção/Planejamento e Construção) e possui cinco processos (Desenvolver um modelo abrangente, Gerar uma lista de funcionalidades, Planejar por funcionalidade, Detalhar por funcionalidade e Construir por funcionalidade). A **figura 2** abaixo ilustrará a descrição dos processos que será posteriormente feita.



Comment [A42]: E a fonte?

Figura 2 - Ciclo de Vida do FDD

Desenvolver um modelo abrangente: essa etapa começa com uma análise superficial do escopo do sistema e seu contexto, seguido de um estudo do(s) domínio(s) de negócio(s) do sistema que leva a criação do referido modelo. Depois é criada uma modelagem superficial para cada área de domínio existente. Os modelos decorrentes das referidas atividades serão revisados por um grupo de membros do projeto e melhorias são colocadas e discutidas. Finalmente, os modelos são fundidos para gerar um modelo geral do domínio do sistema. Dentro desse processo existem algumas sub-atividades:

- Formar equipe de modelagem;
- Estudar o domínio de negócio;
- Estudar os documentos;
- Formar várias equipes pequenas para sugerir uma solução de modelo;
- Desenvolver o modelo escolhido;
- Refinar o modelo geral gerado;
- Escrever notas explicativas sobre o modelo final.

Gerar uma lista de funcionalidades: as práticas e o conhecimento adquirido no processo anterior são essenciais para esta fase. Nesta, será elaborada uma lista de funcionalidades do sistema decompondo as áreas de domínio obtidas. Cada funcionalidade é uma pequena tarefa a ser implementada que agregue valor ao cliente. Devem seguir o formato <ação> <resultado> <objeto>. Como no processo anterior, este também possui sub-atividades:

- Escolher uma equipe para gerar uma lista de funcionalidades;
- Gerar a lista de funcionalidades.

Planejar por funcionalidade: é feito o planejamento de desenvolvimento de cada funcionalidade da lista obtida da fase anterior. Os programadores-chefe recebem classes ou trechos de código e serão responsáveis pelos mesmos. As sub-atividades desse processo são:

- Formar uma equipe de planejamento;
- Determinar a sequência de desenvolvimento das funcionalidades;
- Designar atividades de negócio para os programadores-chefe; e
- Designar classes para os desenvolvedores.

Detalhar por funcionalidade: esse processo requer uma interação entre os programadores-chefe e os proprietários de código, quando da escolha de algumas funcionalidades para que sejam feitos os diagramas de sequência e a modelagem completa das funcionalidades. Devemos observar que não se trata da mesma modelagem do primeiro processo, nesse momento a modelagem é feita para a funcionalidade em questão. Nessa fase o nível de detalhamento é bem maior, pois se deve pensar em classes, métodos e atributos que irão existir. Ao final, é realizada uma inspeção do modelo pela equipe que a fez ou por outra designada. As sub-atividades particulares a esse processo são:

- Formar uma equipe para a funcionalidade em questão;
- Estudar a funcionalidade como parte inserida no modelo de domínio;
- Estudar documentos relacionados á funcionalidade;
- Desenvolver diagrama de sequência;
- Refinar objeto modelo;
- Escrever as classes e as assinaturas dos métodos (tipo de retorno, parâmetros e exceções lançadas);
- Realizar inspeção da modelagem.

Construir por funcionalidade: substanciado pelo processo anterior, o programador-chefe designa um programador para desenvolver o código e finalmente ele passa a ser criado, os testes escritos e a funcionalidade ganha vida. Nesse momento temos as sub-atividades em execução, a saber:

- Implementar as regras de negócio das classes;
- Inspeccionar código;
- Conduzir testes unitários; e
- Release da funcionalidade.

Os três primeiros processos citados, que compõem a primeira fase, acontecem de forma sequenciada, lembrando os métodos tradicionais, porém os processos estão em uma mesma fase. Já os dois últimos, que estão na

segunda fase do FDD, possuem uma dinâmica interativa e incremental. Isso reforça a afirmação feita por Michele Sliger no início da abordagem sobre FDD.

2.7. Tópicos de Pesquisa

Combinação de Processos Ágeis de Desenvolvimento de Software: A literatura e a prática diária mostram que algumas organizações, projetos e/ou times de desenvolvimento não conseguem aplicar fielmente todas as recomendações de determinado PADS. Com isso passam a combinar práticas de mais de um processo ágil ou juntá-los em sua completude para que o mesmo aconteça.

Desenvolvimento Distribuído de Software (DDS) com Metodologias Ágeis: O DDS é uma necessidade inquestionável nos dias atuais quando se buscam fatores como qualidade, competitividade, redução de custos, mão-de-obra qualificada entre outros. Porém existem alguns desafios que precisam ser combatidos/vencidos, os quais são categorizá-los em: pessoas, processos, tecnologia, gestão e, principalmente, comunicação. Ao tempo em que o estudo/uso de metodologias ágeis é crescente, seus resultados são muito positivos e estão preenchendo as lacunas deixadas pelos processos tradicionais. Frente a isso, demanda-se um casamento entre as duas abordagens de forma a se ter processos adaptados e/ou a criação de um único processo para atender a problemática do DDS.

Aproximação dos Processos Ágeis de Desenvolvimento de Software e dos Modelos de Qualidade de Software (MQS): A literatura nos oferece experiências da aproximação de processos ágeis e dos modelos de qualidade de software, porém embora existam diversas adequações do uso do primeiro no segundo, não acontece de forma completa, ou seja, nem toda a área de processo CMMI ou MPS.BR é satisfeito somente com a utilização de abordagens ágeis.

2.8. Sugestões de Leitura

- Para ampliar o entendimento sobre Extreme Programming é recomendado à leitura dos livros Programação Extrema (XP) Explicada: Acolha as Mudanças. Bookman, 2004, Kent Beck (versão original em inglês) e Extreme Programming. Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. Novatec, 2004, Vinícius Malhões Teles.
- Para conhecer detalhadamente o Scrum é recomendável a leitura do Guia do Scrum, disponível em vários idiomas, inclusive em português, no site <http://www.scrumalliance.org>. Nesse endereço é possível encontrar outros artigos e recursos relacionados ao Scrum.
- Para agregar conhecimento em Feature Driven Development (FDD), a partir de um foco prático e adaptativo da mesma, recomendamos visitar o site da empresa Heptagon www.heptagon.com.br, pois além de seu conteúdo próprio, dispõe de inúmeros links para o referido assunto. E para se expandir no assunto a leitura do livro *A practical Guide to Feature Driven Development*. 2002, Stephen Palmer.
- Para aprofundar seus conhecimentos em processos ágeis, visite o endereço <http://agilemanifesto.org>. Nele é encontrado os princípios e valores propostos no Manifesto Ágil, além de disponibilizar detalhes dos autores e links para as suas metodologias.

2.9. Exercícios

1. O que motivou o surgimento de processos ágeis de desenvolvimento?
2. Quais os valores propostos pelo Movimento Ágil? Cite também alguns princípios.
3. A *Agile Modeling* é uma metodologia ágil? Justifique.
4. Compare o XP com o Scrum.
5. Quais os valores propostos por Beck no *Extreme Programming*?
6. Cite e explique as práticas do XP que você considera mais importante.
7. Caracterize a metodologia FDD.
8. O que é Sprint? Quais os papéis que estão envolvidos na Sprint?
9. Comente o ciclo de vida do Scrum.
10. Compare os processos tradicionais de desenvolvimento com os ágeis.

Referências

- ABRAHAMSSON, P., WARSTA, J., SIPONEN, M.T., RONKAINEN, J. (2003) *New Directions on Agile Methods: A Comparative Analysis*. In: ICSE 2003, USA.
- AGILE MANIFESTO (2001). <http://www.agilemanifesto.org>. Acesso em: setembro 2009.
- AMBLER, S. (2002) *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. New York: Wiley Computer Publishing.
- AMBLER, S. (2009) *Agile Modeling*. Disponível em: <http://www.agilemodeling.com>. Acesso em: novembro 2009.
- ANDERSON, D. J. (2003) *Agile Management for Software Engineering: Using the Theory of Constraints for Business Results*.
- BECK, Kent; FOWLER, Martin. (2001) *Planning Extreme Programming*. 1ª edição. Boston: Addison-Wesley.
- BECK, Kent. (1999) *Extreme Programming Explained: Embrace Change*. 1ª edição. Boston: Addison-Wesley.
- BECK, Kent; ANDRES, Cynthia. (2004) *Extreme Programming Explained: Embrace Change*. 2ª edição. Boston: Addison-Wesley.
- COAD, Peter; LEFEBVRE, Eric; LUCA, Jeff. (1999) *Java Modeling In Color With UML: Enterprise Components and Process*. Upper Saddle River, N.J.: Prentice Hall.
- COCKBURN, A. (2002). *Agile Software Development*. Boston: Addison-Wesley.
- FOWLER, M. (2009) *The New Methodology*. Disponível em: <http://martinfowler.com/articles/newMethodology.html>. Acesso em: setembro 2009.
- HIGSMITH, J. (2002) *Agile software development ecosystems*. Boston, MA., Pearson Education.
- HIGSMITH, J. (2004) *Agile Project Management - Creating Innovative Products*, Addison-Wesley.
- LARMAN, Craig. (2003) *Agile and iterative development: a manager's guide*. Addison-Wesley.
- KOSCIANSKI, A., SOARES, M. (2006) *Qualidade de Software*. 2ª edição. São Paulo: Novatec.

- PALMER S. R., FELSING J. M. (2002) *A Practical Guide to Feature-Driven Development (The Coad Series)*. Prentice Hall PTR, USA.
- SLIGER, Michele; BRODERICK, Stacia. (2008) *The Software Project Manager's Bridge to Agility*. Addison Wesley Professional, 2008.
- SCHWABER, K. (2008) *Agile Project Management with Scrum*. Redmond: Microsoft Press.
- SCHWABER, K., BEEDLE, M. (2001) *Agile Software Development with Scrum*. City: Prentice Hall.
- SCHWABER, K. (2009) Guia do Scrum. Disponível em: <http://www.scrumalliance.org/resources>.
- SZALVAY, V. (2007) Glossary of Scrum Terms. Disponível em: <http://www.scrumalliance.org/articles/39-glossary-of-scrum-terms#1117>
- STAPLETON, J. (2003). DSDM, *Business Focused Development*. Addison-Wesley.
- SATO, D. (2007) Uso eficaz de métricas em desenvolvimento de software. 155 p. Dissertação (Mestrado em Ciência da Computação). Instituto de Matemática e Estatística – Universidade de São Paulo, São Paulo, 2007.
- TELES, Vinícius Manhães. *Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade*. Novatec. 2004
- VERSIONONE (2008). 3° Annual Survey: *The State of Agile Development*. Disponível em: http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf.

Índice

3.1 INTRODUÇÃO.....	2
3.2 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE.....	3
3.2.1 MOTIVAÇÕES PARA O DDS	3
ÍNDICE 3.2.1 REPETIDO	
3.2.1 NÍVEIS DE DISPERSÃO	4
3.2.2 MODELOS DE NEGÓCIO	5
3.2.3 DESAFIOS	6
3.3 PROCESSOS PARA DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE.....	7
3.4 PROCESSOS E ADAPTAÇÃO DAS PRÁTICAS EM PROJETOS DDS.....	7
3.4.1 MODELO DE KAROLAK [1998]	8
3.4.2 USO DE PRÁTICAS ÁGEIS	10
3.4.2.1 DXP – DISTRIBUTED EXTREME PROGRAMMING	11
3.4.2.2 ADOÇÃO DE SCRUM EM UM AMBIENTE DDS	12
3.5 TÓPICOS DE PESQUISA	15
SEÇÃO CONSIDERAÇÕES FINAIS DEVE VIR ANTES DE TOPICOS DE PESQUISA	
3.6 CONSIDERAÇÕES FINAIS	15
3.7 EXERCÍCIOS.....	16
SEÇÃO EXERCÍCIOS DEVE SER APÓS SUGESTÕES DE LEITURA E ANTES DE REFERÊNCIAS	
3.8 SUGESTÕES DE LEITURA	16
REFERÊNCIAS.....	18

Comment [A43]: Seria bom por uma lista de figures/tabelas.

Capítulo

3

Processos para desenvolvimento distribuído de Software

Camila Cunha Borges

O objetivo do capítulo é apresentar como os modelos de processos e práticas de desenvolvimento de software podem ser aplicados em um ambiente de desenvolvimento distribuído de software.

Comment [A44]:

1. Na formatação de todos os parágrafos do capítulo, o espaçamento entre linhas deve ser de "simples" e não "1,5 linha"
2. Na formatação de todos os parágrafos, o espaçamento antes é de 6 pontos

3.1 Introdução

Nas últimas quatro décadas podemos observar que o software passou a ser o elemento-chave da evolução de sistemas e produtos baseados em computador [PRESSMAN 2007], onde é necessário desenvolver software com rapidez e qualidade. É importante observar que, à medida que o tempo passa a forma de se desenvolver um software vem passando por mudanças e os problemas relativos ao desenvolvimento continuam semelhantes: usuários insatisfeitos, longo tempo de desenvolvimento, nível de qualidade, dificuldade de comunicação, etc.

Comment [A45]: Tamanho da fonte deve ser 13

Comment [A46]: Virgula após a palavra passa: locução adverbial temporal. O ideal seria substituir essa expressão por outra para não haver repetição do termos "que". Ex: "com o passar do tempo"

Com a globalização dos negócios, surgem grandes desafios para o processo desenvolvimento de software, que está cada vez mais distribuído e global [AUDY 2008]. Sendo assim, observamos a necessidade de implantação de métodos e ferramentas para a melhoria do processo de desenvolvimento distribuído de

Comment [A47]:

Não está com a tabulação correta de 1,27 cm.
Ajustar tabulação padrão para o capítulo para esse valor 1,27 (ver template SBC)

3.2 Desenvolvimento Distribuído de Software

Grandes investimentos têm permitido uma movimentação do mercado local para o global, assim são criadas novas formas de colaboração e competição na área de Engenharia de Software [DAMIAN 2006]. Neste ambiente, muitas organizações encontraram no Desenvolvimento Distribuído de Software (DDS) uma chance para obter sucesso nos negócios.

Segundo AUDY e PRIKLADNICKI, [2008], o DDS ganha cada vez mais força, motivado por três fatores ligados ao ambiente de negócios: (1) a globalização, (2) o crescimento da importância dos sistemas de informação nas empresas e (3) os processos de terceirização que geram um ambiente propício a esse cenário de desenvolvimento.

CARMEL [1999] afirma que as principais características que diferenciam o desenvolvimento co-localizado do desenvolvimento distribuído são: distância, diferenças de fuso horário e diferenças culturais. Distância refere-se à distribuição geográfica dos desenvolvedores e clientes finais. Diferenças culturais idioma, tradições, costumes, comportamentos e normas locais.

De acordo com PRIKLADNICKI [2003], o desenvolvimento distribuído criou uma nova classe de problemas a serem resolvidos pelos pesquisadores na área de desenvolvimento de software. Estas mudanças impactam não apenas no mercado propriamente dito, mas também na maneira como os produtos são criados, modelados, construídos, testados e entregues aos clientes.

3.2.1 Motivações para o DDS

O desenvolvimento de Software era realizado por pessoas com alto grau de especialização, trabalhando em centros de processamento de dados (CPD) em países avançados. Atualmente, o desenvolvimento de *software* vem ocorrendo de uma forma cada vez mais distribuída. [PRIKLADNICKI 2003].

As organizações visam obter vantagens competitivas associadas ao custo, qualidade e flexibilidade no desenvolvimento de *software*. Na maioria dos casos esse processo ocorre no mesmo país ou em regiões com incentivos fiscais. Algumas empresas buscam soluções em outros países (soluções globais), assim obtendo maiores vantagens competitivas.

Segundo a *International Data Group* (IDC) [2006] pode-se ter uma economia entre 25% e 50% em termos de custo quando grandes projetos são transferidos para operações *offshore* (em outro país). Além da redução de custo, é observada a disponibilidade de profissionais habilitados para trabalhar em outro idioma e incentivos de governos locais contribuem para

Comment [A48]: Tamanho da fonte deve ser 13
Idem para todas as outras subseções de segundo nível do capítulo : 3.3, 3.4, 3.5, 3.6 ,ETC.

Comment [A49]: Padrão de referência incorreto

Comment [A50]: Em vez de usar (1), (2),(3), fica melhor colocar tópicos:
• A
• O crescimento
• Os processos

Comment [A51]: Padrão de referência incorreto

Comment [A52]: Não há hífen : colocalizado

Comment [A53]: Grafia incorreta

Comment [A54]: Padrão de referência incorreto

Comment [A55]: Tamanho da fonte deve ser 12
Idem para todas as outras subseções de terceiro nível em diante do capítulo; Ex. 3.2.1 , 3.2.2 , 3.2.3 , 3.4.2 , 3.4.2..1 , ETC

Comment [A56]:
Formatação em itálico incorreta
Observar esse termo *software* e outros, ao longo do capítulo que não podem estar ser escritos em itálico.

Comment [A57]: Padrão de referência incorreto

Existem diversas razões para a aplicação do DDS, em seguida são listadas algumas:

- **Demanda e custo:** Com o aumento na demanda por profissionais de software, o custo da mão-de-obra sofreu um aumento conforme as organizações competiam por suas contratações [KAROLAK 1998]. Assim a disponibilidade de recursos equivalentes em outras localidades tornou-se um grande atrativo.
- **Rapidez de resposta ao mercado:** A possibilidade de um desenvolvimento *follow-the-sun*, onde existem 24h contínuas, é um grande atrativo para empresas que visam reduzir o *time-to-market* (tempo para colocar o produto no mercado).
- **Mercado e presença global:** À medida de os custos são reduzidos e há um aumento no poder computacional, o mercado global de informática cresce. Assim o DDS é uma opção para atender a demanda do mercado global.

Comment [A58]: O termo é existem ou exigem ??

3.2.1 Níveis de Dispersão

O nível de dispersão dos atores envolvidos em um processo é uma característica importante do DDS. O entendimento dos níveis de dispersão auxilia na identificação de possíveis dificuldades. Se existe uma distancia de 30 metros ou mais entre os colaboradores, a frequência da comunicação diminui na mesma proporção aos colaboradores distribuídos a milhares de metros [HERBSLEB 2001]. É importante entender o nível de distancias e suas implicações para as equipes. A seguir apresentamos tipos de distância física (Figura 3.1) e principais características:

Comment [A59]: Distância está sem acento em todas as ocorrências do capítulo.

- **Mesma Localização Física:** A empresa possui todos os atores em um mesmo lugar. Nesta situação, não existem dificuldades de reuniões e há uma interação presente entre membros das equipes. Além disso, não há diferença de fuso-horário e/ou diferenças culturais. Neste cenário as dificuldades são as já existentes no desenvolvimento centralizado.
- **Distancia Nacional:** Os grupos de atores estão localizados em um mesmo país, podendo reunir-se em curtos intervalos de tempo. Em alguns países podem ocorrer diferenças no fuso-horário.
- **Distancia Continental:** As equipes de atores estão localizadas em países diferentes, necessariamente no mesmo continente. As reuniões são mais difíceis de acontecerem face a face. Além disso, o fuso-horário dificulta as interações entre os

- **Distancia Global:** Os grupos de atores estão em países e continentes diferentes, formando distribuição global. A comunicação e diferenças culturais neste cenário podem ser barreiras para o andamento do projeto e as reuniões face a face geralmente acontecem no início do projeto.

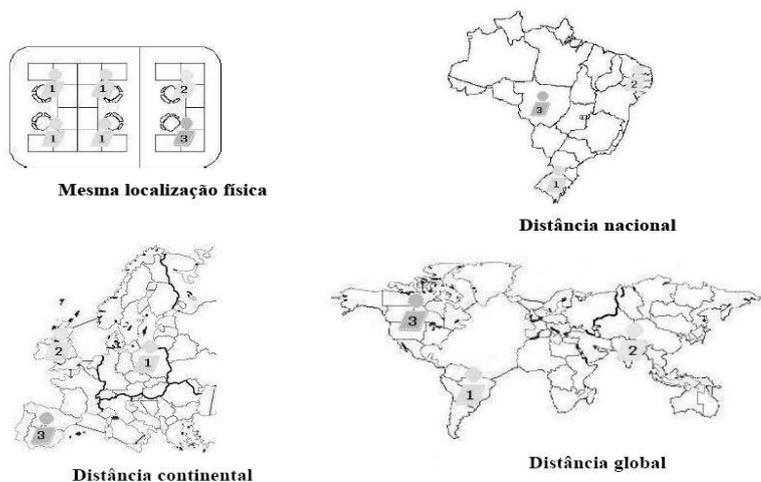


Figura 3.1 – Tipos de Distância Física [PRIKLADINIKI 2007]

3.2.2 Modelos de Negócio

De acordo com PRIKLADINIKI [2007], entre as relações existentes entre clientes e provedores de serviço, podemos caracterizar *Outsourcing* (terceirização) e *Insourcing* (subsidiárias da mesma empresa) como as principais relações existentes. Quanto à dimensão relacionada com a distância geográfica, a distribuição pode ser *Onshore* (em um país diferente) ou *Offshore* (no mesmo país). A seguir, os modelos são definidos:

- **Onshore Insourcing:** Existe um departamento dentro da própria empresa ou uma subsidiária da empresa no mesmo país (*onshore*) que provê serviços de desenvolvimento de software através de projetos internos (*insourcing*).
- **Onshore Outsourcing ou Outsourcing:** É a contratação de uma empresa terceirizada (*Outsourcing*) para o desenvolvimento de determinados serviços ou produtos de software. A empresa terceirizada está localizada no mesmo país da empresa contratante (*onshore*).
- **Offshore Outsourcing ou Offshoring:** É a contratação de uma empresa terceirizada

Comment [A60]:

Fonte de figuras/tabelas deve ser Helvética 10. IDEM para figuras 3.2, 3.3, 3.4 e 3.5 .

SUGESTÃO: Colocar após o Índice, uma Lista de Figuras

Comment [A61]:

Padrão de referência incorreto

software, sendo que ela está localizada em um país diferente da contratante (*offshoring*).

- **Offshore Insourcing ou Internal Offshoring:** É a criação de uma subsidiária da empresa para prover serviços de desenvolvimento de software (*Insourcing*) em um país diferente da empresa contratante (*offshore*).

É importante que seja observado que, além de outras formas de relacionamento entre empresas, também podem surgir outros tipos de distribuição geográfica, resultando em outros tipos de modelos de negócio. Neste livro serão abordados apenas os modelos de negócio já citados na seção 3.2.2.

Comment [A62]: Se já está na seção, basta dizer “nesta seção”

3.2.3 Desafios

Conforme apresentado na seção 3.2.2, o DDS apresenta níveis de dispersão física, distância temporal e diferenças culturais, com isso alguns desafios foram acrescentados ao processo.

O ambiente global apresenta grande impacto na forma como os produtos são concebidos, desenvolvidos, testados e entregues aos clientes [AUDY 2008]. Diferentes tecnologias e características são necessárias para o suporte ao DDS. Entre muitos desafios relacionados ao DDS, nesta seção vamos detalhar desafios focados no processo de desenvolvimento.

- **Arquitetura do Software:** É um dos fatores mais utilizados para a diminuição do esforço entre as equipes. Conforme KAROLAK [1998], uma arquitetura apropriada para o DDS deve se basear no princípio da modularidade, pois permite alocar tarefas complexas de forma distribuída. Com isso há uma redução na complexidade e é permitido um desenvolvimento em paralelo simplificado.
- **Engenharia de Requisitos:** A engenharia de requisitos contém diversas tarefas que necessitam de alto nível de comunicação e coordenação. Com isso os problemas apresentados são mais complexos em um contexto de DDS.
- **Gerência de Configuração:** O gerenciamento de configuração (CM) é a chave para controlar as múltiplas peças em um projeto distribuído. Controlar modificações nos artefatos em cada uma das localidades com o processo de desenvolvimento de todo produto pode ser complexo. Apesar da utilização de

Comment [A63]: Referência fora do padrão

Além disso, o uso de ferramentas de CM compartilhadas por duas ou mais equipes de forma inadequada gera diversos riscos e problemas em projetos DDS.

- **Processo de Desenvolvimento:** Em projetos DDS, o uso de uma metodologia que auxilia a sincronização das atividades é essencial. Com isso todos os membros utilizam uma nomenclatura comum em suas atividades.

Comment [A64]: Não só sincronização, mas sobretudo auxilia na Padronização.

- **Testes:**

Comment [A65]: Sugestão colocar também testes ???

3.3 Processos para Desenvolvimento Distribuído de Software

Em um ambiente de desenvolvimento distribuído, um processo de desenvolvimento comum à equipe é fundamental, tendo em vista que uma metodologia auxilia diretamente na sincronização, fornecendo aos membros da equipe uma nomenclatura comum de tarefas e atividades, e um conjunto comum de expectativas aos elementos envolvidos no processo [PRIKLADNICKI 2008].

A engenharia de software (ES) sempre está apresentando grandes avanços e transformações relacionadas às técnicas, modelos e metodologias. Esses avanços são destacados quando se trabalha com processo de Desenvolvimento Distribuído de Software (DDS), havendo uma necessidade do uso de práticas que dê suporte às dificuldades encontradas nas definições de requisitos que mudam de forma dinâmica no decorrer do tempo. Estudos relacionados a processo para DDS ainda é escasso, sendo assim este capítulo relata o uso de práticas do desenvolvimento tradicional que podem ser implantadas em um ambiente distribuído e as possíveis adaptações.

Comment [A66]: deem

Comment [A67]: Acentuação

3.4 Processos e adaptação das Práticas em projetos DDS

A forma como um produto de software é concebido, desenvolvido, testado e entregue ao cliente sofre grande impacto quando o ambiente de desenvolvimento é distribuído [HERBSLEB 2001]. Assim, a estrutura necessária para o suporte desse tipo de desenvolvimento se diferencia da utilizada em ambientes centralizados. Diferentes características e tecnologias se fazem necessárias, crescendo a importância de alguns

Estratégias, soluções e práticas para tornar esta abordagem um sucesso tornam-se imperativas. O desenvolvimento de ambientes, modelos e ferramentas para gerenciar processos de software neste contexto tornam-se cada vez mais importantes. A seguir apresenta-se uma abordagem relacionada ao processo de desenvolvimento.

3.4.1 Modelo de Karolak

Karolak [1998] aborda o DDS seguindo o ciclo de vida tradicional de um projeto de desenvolvimento de software. O autor propõe um modelo para desenvolver projetos DDS abrangendo as atividades que **dever** ocorrer ao longo do ciclo de vida. A figura abaixo ilustra o modelo proposto (Figura 3.2):

Comment [A68]: Referência fora do padrão

Comment [A69]: devem

Id	Atividades	Engajamento	Requisitos	Modelagem	Implementação	Teste	Entrega	Manutenção
1	Alinhar o negócio	■						
2	Identificar a equipe distribuída	■	■					
3	Identificar as tecnologias	■	■					
4	Definir o contrato	■	■					
5	Dividir o trabalho	■	■	■				
6	Identificar ferramentas e métodos		■	■				
7	Estabelecer responsáveis por SCM		■	■				
8	Identificar e gerenciar riscos	■	■	■	■	■		
9	Controlar a documentação		■	■	■	■	■	■
10	Desenvolver plano e casos de teste			■	■	■		
11	Crear matriz de rastreabilidade		■	■	■	■		
12	Crear matriz de versão de módulos			■	■	■	■	■
13	Crear grupo de manutenção						■	■
14	Controlar a qualidade do software		■	■	■	■	■	■
15	Gerenciar a propriedade intelectual		■	■	■	■		■

Figura 3.2 – Modelo para projetos DDS [KAROLAK 1998]

A seguir apresentam-se as atividades do modelo proposto

Comment [A70]: fica melhor a expressão “serão apresentadas” ou “...serão descritas as definições e objetivos do modelo proposto....”

- **Alinhar o negócio:** Primeira atividade necessária para desenvolver projetos DDS, pois será identificado o tipo de estrutura que será utilizada. Nesta atividade é definido se existirão interações com outras empresas ou se serão criadas unidades da empresa em outras localidades.
- **Identificar a equipe distribuída:** Nesta atividade são definidos os integrantes da equipe, seus respectivos papéis e responsabilidades. A formação da equipe deve considerar os seguintes aspectos: aquisição de confiança, diferenças culturais e relacionamento.
- **Identificar as tecnologias:** Devido à grande demanda de comunicação em projetos DDS, há a necessidade de um apoio tecnológico considerável. Nesta atividade é

identificada a infra-estrutura disponível para os membros das equipes se comunicarem, considerando o nível de dispersão da equipe.

Comment [A71]:
infraestrutura sem hífen de acordo com a nova reforma ortográfica.

- **Definir o contrato:** Um contrato é um documento que define o escopo do que deve ser feito. Quando o projeto é distribuído esta atividade se torna mais complexa.
- **Dividir o trabalho:** Após a identificação da equipe, tecnologia e definição do contrato, é proposta a divisão do esforço de trabalho entre os membros de uma equipe. Deve ser levado em consideração o nível de experiência e a modularidade do projeto.
- **Identificar ferramentas e métodos:** Identificação dos recursos técnicos que serão utilizados na modelagem e implementação do projeto. Deve-se considerar o nível de dispersão da equipe e o processo de desenvolvimento.
- **Estabelecer responsável por SCM:** A gerência de configuração de software (SCM – *Software Configuration Management*) tem como objetivo controlar modificações nos artefatos, dando suporte ao controle de versões. O autor sugere a existência de um grupo responsável pelo controle de configuração e versões do sistema. Por este motivo, esta atividade visa identificar os membros deste grupo, bem como as ferramentas que eles utilizarão e a frequência necessária de reuniões para discutir o andamento do trabalho.
- **Identificar e gerenciar riscos:** Esta atividade faz parte de qualquer projeto. De acordo com o autor, os riscos em projetos DDS tendem a ser mais centrados em aspectos não tão visíveis. Esta atividade deve acontecer em todas as fases do desenvolvimento, exceto entrega e manutenção. Em projetos DDS podem existir três categorias de risco: organizacional, técnico e de comunicação.
- **Controlar a documentação:** É conhecida a resistência em documentar por partes de equipes de desenvolvimento. Em projetos DDS, uma documentação pobre pode causar ineficiência na colaboração. Uma boa documentação pode evitar ambigüidades e facilitar futuras manutenções.
- **Desenvolver plano e casos de teste:** KAROLAK [1998] menciona que um projeto distribuído necessita de pelo menos dois artefatos de teste. O plano de teste com as estratégias, métodos e ambiente documentados e o Caso de teste com as funcionalidades que serão testadas

Comment [A72]: Obs. Há manutenções adaptativas ou evolutivas que demandam de análise de riscos !

Comment [A73]: Sem o trema

- **Criar matriz de rastreabilidade:** uma matriz de rastreabilidade é um artefato que identifica as funcionalidades do projeto e os módulos que as implementam. Este artefato mostra a ligação entre os requisitos e como um requisito influencia em outro.
- **Criar matriz de versão de módulos:** uma matriz de versão de módulos é um artefato que identifica qual versão de um módulo foi utilizada na compilação do código de um projeto. Este artefato é essencial principalmente para a coordenação das atividades e divisão do trabalho entre os membros da equipe do projeto.
- **Criar grupo de manutenção:** O modelo sugere a criação de um grupo responsável por revisar solicitações de alterações após o produto ser entregue ao cliente.
- **Controlar a qualidade do software:** Devem existir atividades que melhoram a qualidade do software a ser desenvolvido, tais como revisões de modelagem, inspeções de código e teste.
- **Gerenciar a propriedade intelectual:** O autor prevê uma atividade onde se busca a devida proteção, levando-se em consideração leis e restrições do local onde o projeto foi desenvolvido (alguns locais fisicamente dispersos podem ter leis muitas vezes desconhecidas pelas organizações).

3.4.2 Uso de Práticas Ágeis

A partir do ano 2000 surgiu uma tendência para o desenvolvimento ágil de aplicações devido a um ritmo acelerado de mudanças e inovações na tecnologia da informação, em organizações e no ambiente de negócios. Desde então vários métodos ágeis foram surgindo, entre eles:

Adaptive Software Development, Crystal, Dynamic Systems Development, eXtreme Programming (XP), Feature Driven Development (FDD) e Scrum.

De acordo com TRAVASSOS [2005], os métodos ágeis são projetados para (1) produzir a primeira entrega em semanas e alcançar feedback rápido e mais cedo; (2) criar soluções mais simples de modo que se houwerem mudanças que haja mais facilidade e menor volume de alterações a serem feitas; (3) melhorar continuamente a qualidade do projeto, fazendo com que a iteração seguinte tenha menor custo de implementação; (4) testar constantemente, para detectar defeitos mais cedo e removê-los com menor custo.

Quando o ambiente é distribuído o uso de em práticas ágeis parece ser incompatível. Práticas ágeis necessitam de comunicação face a face constantemente e a comunicação é um

Comment [A74]: Tirar formato itálico. Pode-se deixar negrito .

Comment [A75]: Referência incorreta

Comment [A76]: houwer

Comment [A77]: se essas partes (1), (2), etc forem colocadas em tópicos, além de se ganhar mais volume de texto, fica mais legível para o leitor.

software devido ao aumento na velocidade de desenvolvimento, alinhamento dos objetivos individuais com os organizacionais e melhoria no desenvolvimento [SUTHERLAND 2007].

3.4.2.1 DXP – *Distributed Extreme Programming*

Conforme abordado no capítulo anterior, a metodologia de desenvolvimento XP (*Extreme Programming*) requer uma comunicação forte e eficaz entre os membros de uma equipe de desenvolvimento de software. Para isso a metodologia enfatiza a necessidade de ter os membros da equipe fisicamente próximos uns dos outros. No entanto, nem sempre os membros da equipe de um projeto estão fisicamente próximos uns dos outros. Nesta seção será apresentada uma adaptação do uso do XP em ambientes DDS "*Distributed Extreme Programming*" (DXP). Estudos mostram que a aplicação do DXP pode ser eficaz e gratificante em projetos cujas equipes estão geograficamente dispersas.

Segundo YOUNG [2008], o DXP aplica princípios XP em um ambiente distribuído, onde os membros das equipes também podem ser altamente móveis. A figura abaixo (Figura 3.3) resume alguns dos aspectos que são relevantes para DXP e alguns que não são referentes ao fato da distribuição ou não das equipes.

Práticas do XP	É necessário o time ser co-localizado?
Jogo do Planejamento Programação em Par Integração Contínua Cliente local	Sim. Estes fatores dependem de uma aproximação entre o negócio, cliente e pessoal técnico.
Releases Pequenos Metáforas Projeto de Software Simples Teste Refatoração Propriedade Coletiva 40 horas semanais Padrão de codificação	Não. Independem se a equipe é Co-localizada ou não.

Figura 3.3 – Adaptação do XP em DDS [Adaptada de KIRCHER 2000]

Conforme apresentado na figura acima, podemos observar que para a utilização do DXP de forma eficaz é necessário que o *Planning Game*, *Pair Programming*, *Continuous Integration* e *On-site Customer* sejam abordadas em uma equipe distribuída. Na figura acima o autor considera que a prática de *Refactoring* não exige um ambiente co-localizado apesar de esta prática exigir o uso da prática *Pair Programming*. Kircher [2000] afirma que estas duas práticas podem iniciar separadamente.

Comment [A78]: Referência incorreta

Comment [A79]: colocalizado

Comment [A80]: colocalizada

Comment [A81]: colocalizado

O DXP assume a existência de algumas condições para que seja eficaz, tais como a disponibilidade de diversas ferramentas e tecnologias. Além das práticas do XP, o DXP assume:

- **Conectividade:** Alguma forma de conectividade precisa existir entre os membros da equipe. Para longas distâncias a *Internet* é utilizada como meio de comunicação.
- **E-Mail:** É um meio de troca de informação muito utilizado no DXP.
- **Gerenciamento de Comunicação:** Para uma gestão eficaz dos artefatos de programação é necessário que seja utilizada uma ferramenta de gerenciamento de configuração.
- **Compartilhamento de Aplicação:** Aplicações ou *Softwares* de compartilhamento de *desktop* devem estar disponíveis para as equipes distribuídas.
- **Uso de Vídeo Conferência:** O uso de áudio e vídeo entre equipes distribuídas é importante para uma comunicação eficaz. Além disso, há o envolvimento do cliente neste meio de comunicação, pois o mesmo não tem disponibilidade de estar no local da reunião.
- **Integração entre os membros de uma equipe móvel:** Caso necessitem se deslocar, podem utilizar equipamentos móveis para participar das atividades de desenvolvimento.

De acordo com YOUNG [2008], o DXP pode integrar membros de equipes remotas e móveis processo de desenvolvimento e, portanto, uma extensão valiosa para o XP tradicional.

Além disso, permite um envolvimento maior com o cliente quando comparado ao XP, principalmente em situações que é necessário ter o cliente *on-site*. O autor enfatiza também a necessidade de atentar para os problemas já existentes em ambientes DDS, tais como comunicação, disponibilidade dos membros das equipes, coordenação, infra-estrutura e gestão.

Comment [A82]: referência incorreta

Comment [A83]: infraestrutura

3.4.2.2 Adoção de *Scrum* em um ambiente DDS

Inserido neste contexto de desenvolvimento distribuído de software, esta seção apresenta a aplicação da metodologia *Scrum*, abordada no capítulo anterior, no processo de desenvolvimento de uma fábrica de software em um ambiente de desenvolvimento distribuído.

A experiência que será descrita nesta seção foi parte da disciplina de Engenharia de *Software* [2009] com um estudo em fábricas de software, fazendo uso de DDS e metodologias

Comment [A84]: referenciar corretamente

de 2009) para desenvolver o produto conforme definido no início do curso. O projeto relatado, denominado *FireScrum*, é uma ferramenta de gerenciamento de projetos que utiliza a metodologia *Scrum*, cujo objetivo é o de facilitar o uso da referida metodologia em ambientes distribuídos.

O desenvolvimento foi dividido em seis módulos: *Core*, *TaskBoard*, *Planning Poker*, *Test Module*, *BugTracking* e *Desktop Agent*. A fábrica era composta por sessenta alunos distribuídos em seis times, na qual cada time era responsável por um dos módulos citados. Todos os componentes de todos os times realizaram suas atividades de forma distribuída. O módulo que será relatado é o *Bugtracking*, composto por nove estudantes divididos em três estados: seis no estado de Pernambuco (distribuídos em Recife e no interior), dois no estado da Paraíba e um na Bahia.

Para o desenvolvimento do módulo *ugtracking*, o time realizou um estudo entre ferramentas *open source* *Mantis* e *Bugzilla*. Assim, foi possível identificar as vantagens e desvantagens de cada uma para que o módulo fosse desenvolvido de forma diferenciada e inovadora, prezando pela simplicidade e usabilidade. O *Firescrum* foi desenvolvido utilizando a ferramenta *Adobe Flex*, o banco de dados utilizado foi o *Postgree SQL* e para o controle de versão foi utilizado o *SVN*.

O processo de desenvolvimento seguiu a metodologia *Scrum*. As *Sprints* tinham duração de 15 dias. A *Sprint Planning* 1, reunião para definir os itens de *backlog* que seriam atendidos na *sprint*, acontecia de forma presencial semanalmente após a aula. A *Sprint Planning* 2, reunião na qual são definidas as tarefas necessárias à implementação das funcionalidades definidas na *Sprint Planning* 1, acontecia de forma remota utilizando os seguintes recursos: *skype*, *MSN* e a planilha de gerenciamento criada no *Google Docs*. As reuniões diárias (*Daily Scrum Meeting*), com o objetivo de acompanhar a realização das tarefas, inicialmente acontecia com o auxílio do *Skype*, *MSN* e posteriormente foi adotado um grupo de *email*, pois os horários dos membros da equipe eram incompatíveis e nem sempre todos poderiam participar das reuniões no horário marcado. Para os participantes que residiam na mesma cidade acontecia encontros em duplas (uso da prática de programação em pares do XP) para discutir a sobre o desenvolvimento, em seguida as dúvidas e conclusões eram postadas no grupo de *email*.

Ao longo do desenvolvimento, o time manteve sempre evidente e aplicada à filosofia de que cada membro era seu próprio gerente e responsável pelos resultados do projeto. Práticas foram acordadas para que os resultados necessários à conclusão da *sprint* fossem alcançados, aprendizado e bom convívio entre os membros. A principal delas foi que questionamentos

Comment [A85]: BugTracking

Comment [A86]: backlog ?

viriam após a conclusão de qualquer tarefa/, ou seja, cada membro estava focado em concluir tarefas e manter os meios de gerenciamento atualizados.

Os membros acompanhavam a evolução de três artefatos: a planilha de tarefas no *Google Docs*, o *burndown* e o grupo de *email*. Isso possibilitou que o time mantivesse durante todo o processo de desenvolvimento um autogerenciamento satisfatório ao atendimento das *sprints*. A figura abaixo (Figura 3.4) mostra os gráficos das cinco *sprints* do projeto de desenvolvimento e a planilha de gerenciamento do *Google docs* (Figura 3.5).

Figura 3.4 – Gráfico *Burndown*

X

Figura 3.5 – Planilha de Gerenciamento

Comment [A87]: Inserir figures 3.4 e 3.5

Adaptando a metodologia para solução dos problemas

A grande dificuldade foi à realização da *Daily Scrum Meeting* (DSM), pois os integrantes do time possuíam atividades paralelas, tais como outras disciplinas da pós-graduação e alguns até trabalhavam, assim os horários disponíveis eram incompatíveis. De forma geral, estão listados abaixo problemas encontrados e adaptações na metodologia para um melhor resultado.

- Foi determinado pelo time que a DSM seria realizada a cada dois dias, assim evitando o risco de algum membro não ter nada a informar do que foi feito no dia, pois nem todos os membros do time estavam disponíveis todos os dias para executar tarefas relacionadas ao projeto.
- Para realizar a DSM, primeiramente utilizamos o *Skype e MSN* e foi acordado o horário das 20h30min para a reunião remota. Esta foi uma tentativa que não deu certo, pois a reunião tornava-se cansativa a freqüentes eram as perdas na conexão.
- A demora na construção de frases claras é característica de perda de foco nas reuniões, assim as reuniões se prolongavam mais que o necessário e acabássemos entrando em peculiaridades dos problemas.
- Foi criado um grupo de email, e foi acordado que a cada DSM postaríamos até 23h as respostas para as perguntas: O que foi feito até hoje?; O que será feito até a próxima DSM?; e Quais os impedimentos?.

- Foi criada uma planilha no *Google Docs*, na qual constava o *Product Backlog*, o objetivo de cada *sprint* e suas respectivas tarefas, o *burndown* e os impedimentos. Assim era possível acompanhar a dinâmica do time.

Lições aprendidas com o uso do *Scrum* em um ambiente DDS

- Integrantes do time auto-gerenciáveis: A utilização do *Scrum* mostrou que para o sucesso do projeto é indispensável que os participantes sejam auto-gerenciáveis. E apesar da dispersão entre os participantes do time foi possível obter um bom resultado.
- Implementação: A experiência adquirida pela equipe com a utilização do *Scrum* em um ambiente DDS revela que a programação realizada a distancia é possível de ser aplicada com o suporte de ferramentas disponíveis, tais como *emails* ou *MSN*.
- Comunicação: Várias ferramentas foram utilizadas e algumas não atingiam o objetivo do time por não suportar vários usuários conectados ao mesmo tempo, por exemplo.

3.5 Considerações finais

O desenvolvimento de software sempre se apresentou de forma complexa. Existe uma série de problemas e desafios inerentes ao processo. Assim como o processo de desenvolvimento de software tem se tornado cada vez mais complexo, a distribuição das equipes no tempo e no espaço tem tornado os projetos distribuídos cada vez mais comuns. O DDS, ao acrescentar fatores como dispersão geográfica, dispersão temporal e diferenças culturais, acentuaram alguns dos desafios existentes e acrescentou novos desafios ao processo de desenvolvimento. O trabalho em ambientes de DDS é mais complexo do que em ambientes centralizados e não existem métodos ou práticas específicas para o ambiente distribuído.

3.6 Tópicos de Pesquisa

Mecanismos de coordenação em ambientes DDS não existem ou são falhos [Herbsleb 2001]. A distância afeta a colaboração entre as equipes, pois há uma menor frequência de comunicação, comunicação ineficiente, falta de percepção, além da incompatibilidade de processos, ferramentas e práticas de trabalho.

Nos últimos anos muitos trabalhos apresentam propostas de solução para as dificuldades e desafios existentes em projetos DDS. As pesquisas podem se concentrar em questões como o

Comment [A88]:
autogerenciáveis

Comment [A89]: idem

Comment [A90]: letra minuscule e expressão "apesar da dispersão entre os participantes" deve estar entre vírgulas

Comment [A91]: acentuou

Comment [A92]: ?? não existem métodos ou práticas ??
Ficou conflitante com seu trabalho/capítulo.
Talvez você esteja querendo dizer que " não existem práticas maduras ou consolidadas suficientes .et c. ..."

este capítulo aborda processos para DDS, a seguir apresentamos uma lista de tópicos de pesquisa.

- **Processo de desenvolvimento em um ambiente DDS:** A definição de um processo que considere o contexto de uma equipe distribuída. Os modelos de qualidade de software reconhecidos internacionalmente (CMMI) e nacionalmente (MPS-BR) orientam as organizações no desenvolvimento de processos, mas não propõem modelos para distribuição e distância.
- **Uso de práticas em ambientes DDS:** O uso de uma prática pode ser aplicado em diferentes modelos de negócio de um ambiente DDS? Podemos comparar o modelo *Outsourcing* e o *Inourcing*.
- **Ferramentas de colaboração:** Atualmente existem muitas ferramentas que oferecem suporte as atividades do ciclo de vida do desenvolvimento de um *software*. Estas ferramentas são adaptadas para o cenário distribuído. Neste contexto, observa-se a necessidade de ferramentas que oferecem suporte ao *awareness* de atividade (quem está fazendo o quê), de processo (quem deve fazer o quê) e de disponibilidade (quem está disponível quando).

3.7 Sugestões de Leitura

- Desenvolvimento Distribuído de Software [Prikladiniki 2008];
- Global Software Teams [Carmel 1999];
- MuNDDoS: Um Modelo de Referência para Desenvolvimento Distribuído de Software [Prikladiniki 2003];
- Distributed Scrum: Agile Project Management with Outsourced Development Teams [Sutherland 2007].

Comment [A93]: Sugestões de leitura são para aprofundar determinados assuntos que foram tratados superficialmente no capítulo ou para assuntos/temas relevantes que não foram vistos no capítulo. Assim, Para cada tópico de sugestão listado, descrever mais o que é (o problema) e a razão do tópico. Além disso, Não se deve usar referências bibliográficas nessa seção, e sim, o nome, autor, editora do livro sugerido ou link a ser pesquisado.

3.8 Exercícios

1. Defina o que é Desenvolvimento Distribuído de Software.
2. Quais as vantagens que uma organização tem ao utilizar um processo DDS?
3. Quais são os níveis de dispersão em um ambiente DDS? Exemplifique.
4. Quais os modelos de negócio em um ambiente DDS? Exemplifique.
5. Quais as principais dificuldades ao realizar um projeto DDS?

7. Existem empresas/organizações que utilizam somente DDS em seus projetos?

Cite exemplos?

Comment [A94]: Sugestão:
Adicionei mais dois exercícios.
O ideal é ter uma média de 10 exercícios por capítulo

Referências

Herbsleb, J. D., Moitra, D. "Global Software Development", IEEE Software, March/April, EUA, 2001, p. 16-20.

Karolak, D. W. "Global Software Development – Managing Virtual Teams and Environments". Los Alamitos, IEEE Computer Society, EUA, 1998, 159p.

Kiel, L. "Experiences in Distributed Development: A Case Study", In: Workshop on Global Software Development at ICSE, Oregon, EUA, 2003, 4p.

Kircher, M., Jain, P., Levine, A. "Distributed Extreme Programming", IEEE, Agile 2008.

Herbsleb, J.D., Mockus, A., Finholt, T.A. e Grinter, R. E. "An empirical study of global software development: distance and speed", In: ICSE 2001, Toronto, Canada.

Carmel, E. "Global Software Teams – Collaborating Across Borders and Time-Zones" Prentice Hall, EUA, 1999, 269p.

Marquardt, M. J., Horvath, L. "Global Teams: how top multinationals span boundaries and cultures with high-speed teamwork". Davies-Black. Palo Alto, EUA, 2001.

Young, C., Terashima, H. "How Did We Adapt Agile Processes to Our Distributed Development?", IEEE, Agile 2008.

Prikladnicki, R., Audy, J. L. N., Evaristo, R. "Global Software Development in Practice: Lessons Learned", Journal of Software Process: Practice and Improvement – Special Issue on Global Software Development, 2004.

Prikladnicki, R. "MuNDDoS: Um Modelo de Referência para Desenvolvimento Distribuído de Software". Dissertação de Mestrado, PPGCC – PUCRS, Brasil, 2003.

J. L. N. Prikladinicki, R.; Audy. Desenvolvimento Distribuído de Software. 2007.

Perrelli, Hermano. Visão Geral do RUP. Centro de Informática, Universidade Federal de Pernambuco. Disponível em: <http://www.cin.ufpe.br/~if717/slides/3-visao-geral-do-rup.pdf>. Acessado em 20 Maio 2009.

PRESSMAN, Roger S. Software Engineering: a practitioner's approach. EUA: McGraw Hill, 2001. 860 p.

Sutherland, J., “Distributed Scrum: Agile Project Management with Outsourced Development Teams”, HICSS, 2007.

Teles, Vinícius Manhães. Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. 1. ed. São paulo: Novatec, 2004. 320 p.

Travassos, G. H., Abrantes, J. F., “Caracterização de Métodos Ágeis de Desenvolvimento de Software”

Capítulo

10

Comment [A96]: Não seria 4??

Desenvolvimento de Software Dirigido a Modelos

Almir Buarque

O objetivo geral deste capítulo é apresentar o processo desenvolvimento de software dirigido a modelos (MDD), padronizado pela Arquitetura Dirigida a Modelos (MDA) do grupo OMG, sua relevância para elevação da qualidade do processo de engenharia de software e, conseqüentemente, do produto. Duas abordagens MDD serão descritas: OO-Method e AndroMDA. O capítulo mencionará ainda os problemas e desafios atuais do processo de desenvolvimento dirigido por modelos. Será apresentada mais detalhadamente, a abordagem OO-Method por ser uma referência na literatura MDD, ter precisão e definição semântica baseada na linguagem formal orientada a objeto chamada OASIS e por ser totalmente suportado pelo ambiente OLIVANOVA.

Sumário

Comment [A97]: O sumário não seria antes?

10.1 INTRODUÇÃO	3
10.2 ARQUITETURA DIRIGIDA A MODELOS	5
10.2.1. CONCEITOS BÁSICOS	5
10.2.2. PADRÕES OMG E A ARQUITETURA MDA	12
10.3 ABORDAGENS MDD MODELOS	13
10.3.1 OO-METHOD	13
10.3.1.1. O PROCESSO BÁSICO DE TRANSFORMAÇÃO.....	14
10.3.1.2. COMPARAÇÃO COM MDA.....	15
10.3.1.3. O MODELO CONCEITUAL	16
10.3.1.4. O COMPILADOR DE MODELOS	18
10.3.1.5. OLIVANOVA	18
10.3.2 . ANDROMDA	19
10.4 PROBLEMAS E DESAFIOS DOS PROCESSOS MDD	20
10.4.1. VISÃO GERAL	20
10.4.2. LIÇÕES APRENDIDAS NA ADOÇÃO DE SOLUÇÕES MDA.....	21
10.4.3. O PROGRAMA FASTSTART DA OMG	21
10.5. TÓPICOS DE PESQUISA.....	22
10.6 . SUGESTÕES DE LEITURA	22

10.7 . EXERCÍCIOS.....	23
-------------------------------	-----------

REFERÊNCIAS.....	24
-------------------------	-----------

Lista de Figuras

Figura 10.1. Transformações em MDA	7
Figura 10.2 Metamodelo MDA.....	8
Figura 10.3. Transformações de mapeamentos por metamodelos.....	9
Figura 10.4. Transformações com UML Profile.....	11
Figura 10.5. UML Profiles da OMG.....	11
Figura 10.6. Padrões MDA	12
Figura 10.7. Abordagem OO-Method.....	15
Figura 10.8 Diagrama de Classes.....	25
Figura 10.9 Diagrama de Atividades	25

Lista de Tabelas

Tabela 10.1. Comparação do OO-Method com MDA.....	15
---	----

10.1 Introdução

Dentro do contexto de que modelar é uma atividade essencial da engenharia de software, desenvolvimento de software dirigido a modelos "Model Driven Software Development", cujo acrônimo em inglês é MDD, vem representando atualmente um papel central no processo de engenharia de software. Convém lembrar que essa idéia não é nova. Desde a década de 1970, que os métodos formais difundiram o desenvolvimento de software a partir de modelos formais matemáticos e suas transformações até se obter código executável. A partir de um desenvolvimento formal é possível elevar a qualidade do software com técnicas formais de validação e verificação. Com o amadurecimento das linguagens de modelagem de software e a complexidade da conjuntura atual da indústria de software, cada vez mais, essa idéia tem se consolidado através de abordagens que adotam MDD como um padrão de desenvolvimento. Em 2001, quando o grupo OMG especifica a Arquitetura Dirigida a Modelos-MDA (Model Driven Architecture), ele cria uma nova instância de processo de desenvolvimento de software dirigido a modelos (MDD) que já existia há anos, renomeando-a de MDA.

Os principais argumentos para a utilização de um processo de desenvolvimento dirigido a modelos são os seguintes: maior produtividade, portabilidade, interoperabilidade, menor custo, mais facilidade na evolução do software, enfim, maior qualidade do produto. Esses benefícios são evidenciados, por exemplo, num estudo [MDA 2003] que comparou uma produção de software usando-se a tecnologia MDD com o mesmo software fabricado com tecnologia OO tradicional. Isso ocorre principalmente pelas seguintes razões: Primeiramente porque a principal idéia em MDD é a transformação de modelos de maiores níveis de abstração (domínio do problema) em modelos mais concretos (domínio solução) até se obter, por fim, o código do sistema. Depois, o paradigma MDD preconiza que o desenvolvimento inicial e modificações futuras da aplicação sejam efetuados apenas no modelo mais abstrato.

Em processos MDD automatizado, esse modelo abstrato do sistema deve representar com precisão o código, ou seja, ele deve ser executável e ter uma equivalência funcional com todos os outros modelos mais concretos. Dessa forma, as modificações no modelo de mais alto nível de abstração são refletidas automaticamente nos modelos de mais baixo nível, tornando a atividade de modelar no nível mais abstrato o centro de todo processo de desenvolvimento do software e dispensando completamente, nos melhores ambientes MDD, atividades manuais nos modelos de mais baixos níveis de abstração (projeto e implementação).

Entretanto, a indústria de software tem potencializado e exagerado esses benefícios, transmitindo a falsa idéia aos desenvolvedores de que em MDD, apenas com um click ou passo de mágica, obtém-se todas as transformações e o produto de software final. Além disso, passa-se a idéia de que gerar código é o principal objetivo MDD quando é transformar modelos, confundindo os desenvolvedores com várias ferramentas (ambientes) CASE que apenas geram códigos a partir de técnicas diversas e que, na verdade, não transformam modelos. Ademais, existem ainda problemas semânticos, complexidades e imprecisões (ambigüidades) inerentes aos modelos atuais que tornam esse processo de transformação e mapeamentos de modelos uma tarefa árdua e propensa à falhas.

Por outro lado, não há um consenso na comunidade acadêmica sobre qual modelo de maior nível de abstração é mais adequado (necessário e suficiente) para se modelar um sistema, dificultando-se padronizações, interoperabilidade e produzindo-se ambientes MDD que não são integrados com modelos em nível de requisitos que são essenciais para todo o processo de Engenharia de Software. Este capítulo do livro abordará todos esses tópicos referentes ao processo de desenvolvimento de software dirigido a modelos, mas precisamente sobre a arquitetura MDA.

Comment [A98]: Não deveria vir em itálico?

Comment [A99]: Sugiro enumerar as razões

10.2 Arquitetura Dirigida a Modelos

Esta seção objetiva descrever uma visão geral dos padrões OMG, arquitetura MDA e seus conceitos básicos.

10.2.1. Conceitos Básicos

Para um melhor entendimento da arquitetura dirigida a modelos, o padrão MDA do OMG [OMG 2003] define os seguintes conceitos:

- **Modelo**
Um modelo de um sistema é a sua representação (especificação(especificação) funcional, estrutural e comportamental. Uma especificação é dita como formal quando é baseada em uma linguagem que tem uma sintaxe e semântica bem definidas semântica bem definida e, possivelmente, tem também regras de análise, inferência ou prova de seus elementos. Essa sintaxe pode ser gráfica (visual) ou textual. A semântica pode ser mais ou menos formal.
- **Dirigido a Modelos**
MDA é uma abordagem de desenvolvimento de sistema que usa o poder dos modelos. É dirigida a modelos porque provê meios de usar modelos para direcionar o curso de entendimento, projeto, construção, distribuição, operação, manutenção e modificação.
- **Arquitetura**
Arquitetura de um sistema é a especificação de suas partes e conectores, além das regras de interação dessas partes usando os conectores.
- **Ponto de vista (Viewpoint)**
Um ponto de vista de um sistema é uma técnica de abstração, usando um conjunto selecionado de conceitos arquiteturais e regras de estruturação que visa focar ou representar um aspecto (característica) dentro desse sistema. O termo abstração está sendo usado para significar o processo de suprimir (esconder) um detalhe selecionado para estabelecer um modelo simplificado.
- **Plataforma**
Uma plataforma é um conjunto de subsistemas e tecnologias que provê um conjunto coerente de funcionalidade através de interfaces e padrões de uso especificados, que qualquer aplicação (sistema) suportada por essa plataforma pode usar, sem ter que saber os detalhes de como essa funcionalidade provida pela plataforma é implementada.
- **Pontos de Vistas (Modelos) MDA**
 - **Modelo Independente de Computação (CIM)**
É uma visão do sistema a partir de um ponto de vista (viewpoint) independente de computação. O CIM não mostra detalhes da estrutura dos sistemas, sendo usualmente chamado de modelo de domínio ou modelo de negócio negócio e utiliza, em sua especificação, um vocabulário familiar aos usuários do domínio (problema) em questão. Os usuários do CIM geralmente não têm conhecimento sobre modelos ou artefatos usados para realizar as funcionalidades definidas através dos requisitos. Esse modelo foca no ambiente do sistema e nos seus requisitos, deixando os detalhes da estrutura e processamento (computação) do sistema escondidos aos usuários sistema escondido aos usuários (stakeholders) ou, mesmo, esses detalhes são indeterminados.
Dessa forma o CIM tem um importante papel de fazer a ponte (reduzir a lacuna “gap”)”) entre aqueles que são especialistas no domínio do problema e seus requisitos, e aqueles

que são especialistas em projeto (arquitetura) e construção dos artefatos que juntos vão satisfazer aos requisitos do domínio, elicitados pelos usuários.

O CIM é obtido no processo de documentação e especificação dos requisitos, ou seja, ao se especificar um modelo de requisitos para o sistema. Outra forma também de definição do CIM é o modelo de negócios do sistema.

- **Modelo Independente de Plataforma (PIM)**

O PIM foca na operação do sistema (modelo computacional), mas escondendo os detalhes necessários para implantar esse modelo numa plataforma específica. O PIM é único para o sistema e não muda quando se varia de uma plataforma para outra. Esse ponto de vista independente de plataforma pode ser especificado usando-se uma linguagem de modelagem de propósito geral (UML) ou uma linguagem específica (OO-Method) como será visto na seção 10.3.1. O PIM é um modelo conceitual do sistema.

- **Modelo Específico de Plataforma (PSM)**

Este modelo é uma visão do sistema que agrega características e elementos constituintes de uma plataforma específica, contendo informações da tecnologia utilizada na aplicação como a linguagem de programação, os componentes de middleware, a arquitetura de hardware e de software. Para que isso seja possível é necessário o suporte de ferramentas que façam o mapeamento adequado de uma especificação abstrata (PIM) para uma determinada plataforma. O PSM, por sua vez, passa por processo(s) de refinamento(s) para obtenção do nível de especificação desejado. A obtenção desse nível torna possível a transformação do mesmo no código (implementação) da aplicação. O modelo PSM é o responsável por lidar com toda heterogeneidade e complexidade dos diversos tipos de plataformas existentes.

- **Transformações (Mapeamentos)**

A força motriz do padrão MDA é a transformação de modelos, que pode ser realizada entre modelos de um mesmo ponto de vista ou entre pontos de vistas diferentes, tanto num sentido direto quando inverso (reverso). Em qualquer caso, sempre um modelo é usado como parâmetro de entrada para ser transformado em outro modelo.

A figura 10.1 mostra o ciclo (sentido) mais natural MDA, partindo do CIM (modelo de requisitos) até o nível mais baixo de código (implementação).

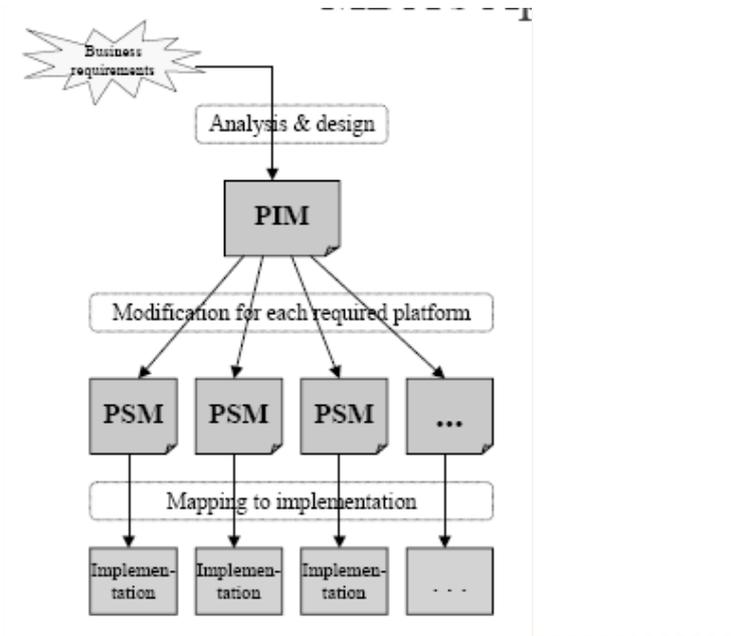


Figura 10.1. Transformações em MDA

Entretanto, é possível também as seguintes transformações é possível também a seguinte transformação (mapeamentos):

- PSM => PIM (Engenharia Reversa)
- PIM => PIM, PSM => PSM (Modelos de mesmo nível (Modelos de mesmo nível)).
- Implementação => PSM (Engenharia Reversa)
- PIM => Implementação Implementação

• **Transformações e Mapeamentos em MDA**

Existe uma quantidade enorme de ferramentas para suportar transformação de modelos. Transformações podem utilizar diferentes técnicas que vão desde uma transformação manual, semi-automática e automatizada. Por exemplo, transformações de PIM para PSM podem ser realizadas através de uso de UML Profiles (extensões UML), uso de padrões (patterns), marcas (markings), metamodelos e transformações automáticas (via algoritmos) [MDA Guide Version 2003]. Os elementos centrais dessas transformações são os mapeamentos dos modelos. Segundo a arquitetura MDA, um mapeamento é um conjunto de regras e técnicas utilizadas para modificar, refinar ou transformar um modelo e se obter um outro modelo. Esse mapeamento, usando-se metamodelos (modelos que descrevem e especificam os modelos originais), facilita a automação. A Figura 10.2 descreve o Metamodelo MDA [MDA 2001]. Observa-se que PIM, PSM e técnicas de mapeamento são baseadas em metamodelos expressos preferencialmente com as tecnologias núcleo do OMG: UML, MOF ou CWM.

Comment [A100]: Fonte??

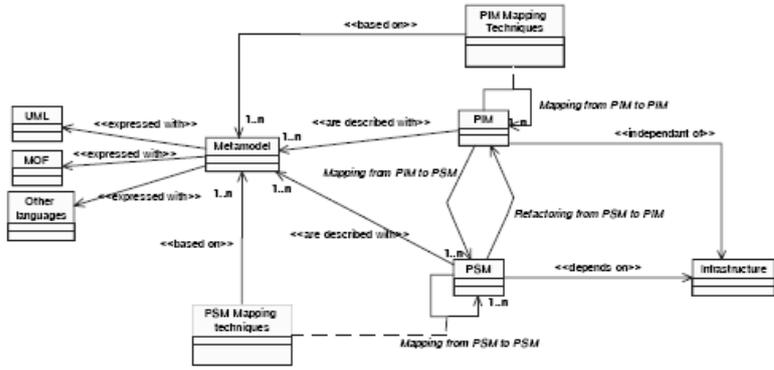


Figura 10.2 Metamodelo MDA

Comment [A101]: Fonte??

Nota-se ainda que o OMG não contempla nesse metamodelo MDA, seu próprio Modelo Independente de Computação (CIM) o que, sob a ótica de Engenharia de Requisitos e Engenharia de Software é como se deixasse uma grande lacuna “gap” a ser preenchida ou, como se o próprio PIM (UML) se unificasse com o CIM (UML), transmitindo a idéia de ser um único modelo capaz de representar de modo completo e consistente todo um sistema.

Para ilustrar o esquema geral de transformações através de metamodelos, considere a figura 10.3 onde um modelo 1 é transformado num modelo 2, usando como entrada do processo o metamodelo A do modelo 1 e produzindo o modelo 2 expresso no metamodelo B. É importante destacar que para realizar essa transformação é necessário ter regras de mapeamento precisas entre esses metamodelos.

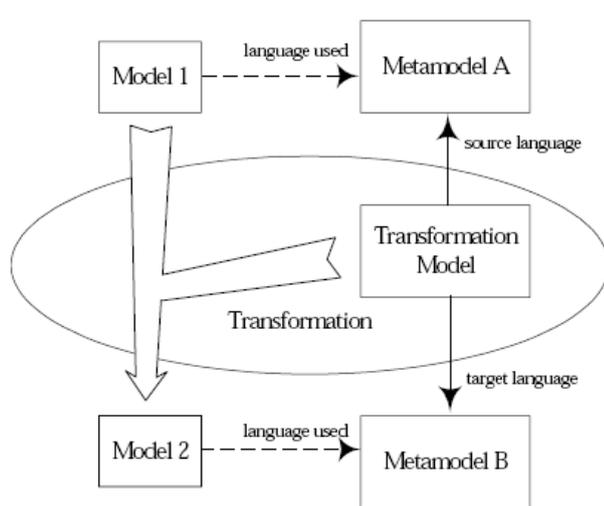


Figura 10.3. Transformações de mapeamentos por metamodelos

De fato, a Linguagem de Modelagem Unificada (UML) é um marco na história de modelagem visual de software, pois antes dela havia várias notações muitas delas incompatíveis entre si. Desde a sua primeira versão (UML 1.0) lançada em 1997, ela recebeu diversas críticas e propostas de extensão. Em 2001, o OMG publicou a UML 2.0. [UML 2.0](#). Alguns dos novos aperfeiçoamentos da UML 2.0 foram:

- Melhor suporte de extensão para outros modelos (linguagens) através do uso de UML Profiles;
- Aperfeiçoamento da expressividade de modelar, incluindo modelagem de processos de negócios, suporte a modelagem de classificadores reusáveis e suporte para modelagem de arquiteturas distribuídas e sistemas heterogêneos;
- Integração com "Actions Semantics" que o desenvolvedor pode usar para definir a semântica de tempo de execução do modelo (aspecto funcional) e prover precisão semântica exigida para analisar modelos e transformá-los em implementações.

Robert B. France em "Model-Driven Development Using UML 2.0: Promises and Pitfalls" [France and Ghosh 2006] cita que padrão UML 2.0 contém um largo conjunto de conceitos de modelagem que são relacionados de um modo complexo. Para cobrir essa complexidade seus projetistas organizaram o padrão UML 2.0 em quatro partes:

- Infra-estrutura: elementos ou construtores básicos da linguagem.
- Super-estrutura: o próprio metamodelo UML.
- Linguagem de Restrição de Objeto (OCL): especificação de consultas, invariantes, restrições e operações em modelos UML.
- Intercâmbio de Diagramas: extensão do metamodelo (Super-estrutura) para dar suporte armazenamento e intercâmbio de informação de modelos UML.

Além de toda essa complexidade, UML carece de precisão semântica, pois muitos dos seus elementos (primitivas) têm diferentes interpretações e varia conforme entendimento do projetista. Isso causa ambigüidades [France and Ghosh 2006]. Também, Oscar Pastor [Pastor and

Comment [A102]: Fonte??

Molina 2007] afirma que a maioria dos métodos baseados em UML tem conceitos como generalização, associação e agregação tão ambíguos e dependentes da interpretação do projetista que o resultado em termos do produto de software é imprevisível associação e agregação tão ambíguas e dependentes da interpretação do projetista que o resultado em termos do produto de software é imprevisível. Isso porque os relacionamentos de classes têm mais semântica do que o proposto por esses métodos. Assim, um modelo conceitual só será preciso se somente esses relacionamentos estão claramente definidos.

Essa imprecisão, aliada da ausência de formalismo de seu metamodelo, faz com que sua validação fique comprometida, e como consequência, erros e inconsistências sejam propagados, durante o refinamento desses elementos, para os níveis de menor abstração da UML [Pastor and Molina 2007].

Para dar um melhor suporte MDD, UML 2.0 lançou o conceito de UML Profile. Esse mecanismo de extensão auxilia a transformação de modelos PIM para PSM específicos, conforme esquema ilustrado na figura 10.4:

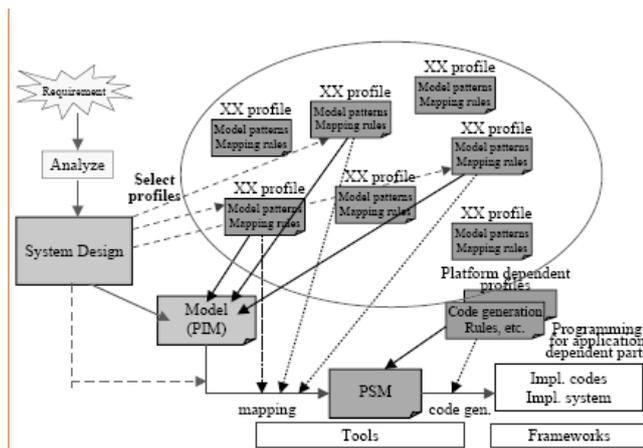


Figura 10.4. Transformações com UML Profile

Atualmente muitas extensões já estão padronizadas pela OMG, algumas estão em processo de padronização e outras ainda em discussão como mostrado na figura 10.5.

Comment [A103]: Fonte??

- | | |
|---|---|
| <ul style="list-style-type: none"> OMG(standardized) - UML Profile for EAI (Enterprise Application Integration) - UML Profile for EDOC (Enterprise Distributed Object Computing) - UML Profile for Schedulability, Performance and Time - UML Profile for CORBA OMG(in process) - UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms - UML for Systems Engineering JCP(standardized) - UML Profile for EJB (JCP) Others (discussing topics, rumor) - UML Profile for WSDL - UML Profile for XML Schema - UML Profile for Persistence Model - UML Profile for Reverse Engineering - UML Profile for Framework Architectures - UML Profile for DCL - UML Profile for Business Modeling - UML profile for Business Analysis | <ul style="list-style-type: none"> - CCA (Component Collaboration Architecture) - Entities Profile - Events Profile - Business Process Profile - Relationship Profile - UML Profile for .NET - UML profile for Interaction design - UML Profile for Database Design - UML profile for hypermedia - UML for Ontology Development - UML profile for DAML - UML Profile for Web applications |
|---|---|

Figura 10.5. UML Profiles da OMG

Comment [A104]: Fonte??

Enfim, devido sua imprecisão semântica e complexidade, UML 2.0 torna-se um problema não somente para desenvolvedores de ferramentas MDD, mas também para os próprios projetistas (grupos de trabalho) da OMG na evolução do padrão [France and Ghosh 2006]. Isso não significa subestimar o valor inegável da UML no contexto da Engenharia de Software, entretanto, afirmar que UML vai ser mesmo o futuro do desenvolvimento de software dirigido por modelos (MDD) só o tempo dirá.

10.2.2. Padrões OMG e a Arquitetura MDA

O surgimento da arquitetura MDA em 2001 foi resultado da necessidade cada vez mais emergente de realizar manutenções em aplicações, integrá-las com outros sistemas, mudar suas infra-estruturas, alterar seus requisitos e lidar com a frequente evolução e criação de novas tecnologias. Além disso, MDA objetiva proporcionar os seguintes benefícios: produtividade, portabilidade, interoperabilidade [interoperabilidade](#)

Para atingir esses objetivos e separar os níveis de abstrações, MDA [OMG 2003] foi definida pela OMG em três camadas conforme figura 10.6, tendo, na primeira camada de especificação (núcleo da arquitetura), padrões que ditam um conjunto de regras para estruturação da especificação expressa nos modelos e que não abordam características de plataformas específicas.

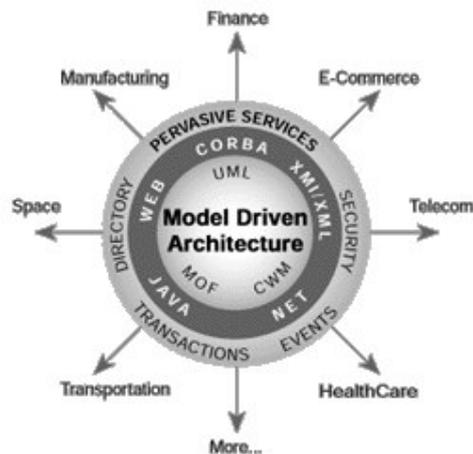


Figura 10.6. Padrões MDA

Esta camada representa o mundo (modelo) PIM. Os padrões também constituem definições propostas pela OMG. São eles:

- Unified Modeling Language (UML): padrão que define uma linguagem de modelagem geral orientada a objetos para especificação, construção e documentação de artefatos de sistemas complexos de software;
- Common Warehouse Metamodel (CWM): padrão para armazenamento de dados que permite fácil manipulação dos mesmos entre ferramentas e plataformas de armazenamento em ambientes heterogêneos distribuídos;
- Meta Object Facility (MOF): padrão que define uma linguagem abstrata para definição de linguagens de modelagem (metamodelos). Ela é utilizada para descrever modelos da UML, CWM e do próprio MOF, além de definir o formato de intercâmbio para modelos, base do padrão XMI (XML Metadata Interchange);

Na segunda camada, encontram-se os modelos PSM que possuem características próprias a determinadas tecnologias e plataformas. Entre elas, algumas seguem padronização da OMG, elevando a resolução dos problemas de integração através da definição de especificações voltadas para interoperabilidade, que sejam abertas e independentes de fornecedores ou fabricantes específicos. São elas:

- XML Metadata Interchange (XMI): padrão para o intercâmbio de modelos através do mapeamento da linguagem definida pelo padrão MOF para o padrão XML do World Wide Web Consortium (W3C);
- Common Object Request Broker Architecture (CORBA): arquitetura que estabelece e simplifica a troca de dados entre sistemas distribuídos.

Na camada PSM, pode-se ter também outros padrões como JAVA EJB, Microsoft .[Microsoft.NET](#), etc.

Na camada mais externa, são exibidos os serviços que a maioria dos domínios de aplicações necessita, para então, serem apresentados os múltiplos domínios que fazem uso desses serviços. Esses serviços podem ser de segurança, persistência, controle de transações, tratamentos de eventos, etc.

Comment [A105]: Fonte??

10.3 Abordagens MDD Modelos

Esta seção tem como principal objetivo descrever a abordagem MDD, chamada OO-Method, que apresenta características de um real ambiente MDD através de uma completa transformação de modelos. O OO-Method inova com o conceito de compilador de modelos “model compiler”, que de fato, é uma máquina virtual de transformação de modelos. Além disso, o modelo de alto nível, chamado modelo conceitual, do OO-Method tem todos seus elementos (primitivas) descritos numa notação visual (gráfica) e que são especificados numa linguagem formal orientada a objeto (OASIS). Essas características fazem com que o OO-Method tenha precisão sintática e semântica suficientes para prover um ambiente capaz, inclusive, de fazer validação de modelos e consequentemente gerar um produto de software final de qualidade.

Nesta seção, será mencionada a ferramenta que implementa OO-Method, chamada OLIVANOVA, e sua comparação com outras ferramentas proprietárias.

Por fim, para não deixar de mencionar o poderoso mundo Open Source em expansão, esta seção também citará uma outra abordagem MDD não proprietária que está se tornando bastante popular: AndroMDA.

10.3.1 OO-Method

A primeira versão do OO-Method foi introduzida em 1992 através da tese de PhD de Oscar Pastor, juntamente com a da linguagem formal, de especificação de sistemas de informação – OASIS [Pastor and Molina 2007]. Deste então, o método incorporou um número de componentes até chegar a versão apresentada neste trabalho. Segundo autor [Pastor and Molina 2007], o método cobre todas as fases do processo de desenvolvimento de software, das fases iniciais de obtenção de requisitos e representação, passando pelo desenvolvimento correspondente do esquema conceitual OO, mais a geração do produto final de software numa plataforma específica. O centro do desenvolvimento do software dirigido por modelos do OO-Method é o Esquema (Modelo) Conceitual que tem como leitmotiv a seguinte afirmação do Prof. Antoni Olivé (Olivé 2005) [Pastor and Molina 2007]:

“Para desenvolver um sistema de informação é necessário e suficiente definir seu esquema conceitual conceitual”

Comment [A106]: Quem disse?

Esta idéia aparece em trabalhos e propostas de alguns pesquisadores de prestígio. Toni Morgan, defende a idéia de usar “Extreme Non-Programing” [Morgan 2002] como argumento de que a principal atividade no desenvolvimento de software é modelagem, e não programação, pois modelagem está no espaço do problema enquanto programar está no espaço da solução. O objetivo final é tornar verdadeira a sentença “O Modelo é o Código”, em vez de “O Código é o Modelo”. Tudo isso é possível se obter, quando se tem um Modelo Conceitual Executável que abstrai de modo completo e consistente todos os aspectos estáticos, dinâmicos e de interação (interface usuário) de um sistema, tal como o do OO-Method, passível de transformação através de um compilador de Esquema Conceitual.

10.3.1.1. O processo básico de transformação

OO-Method estabelece uma distinção clara entre o espaço do problema, onde está definido o esquema conceitual, e o espaço da solução definidos o esquema conceitual, e o espaço da solução, onde é obtido o produto de software que representa o esquema conceitual. Na figura 10.7, o processo se inicia com uma a entrada que representa os requisitos do sistema, não importando por quais processos de engenharia de requisitos esses requisitos foram obtidos, nem o modelo de requisitos utilizado. De forma que esses requisitos são insumos para se criar (projetar) o esquema conceitual. Especificados os quatro modelos que compõe o esquema conceitual: modelo objeto, funcional, dinâmico e de apresentação, é gerado um repositório para conter todos os elementos (primitivas) especificados nos modelos desse esquema conceitual, utilizando-se a linguagem formal orientada objeto OASIS, conforme figura 10.7. Regras de mapeamentos das primitivas desse esquema conceitual para um modelo de aplicação específico de cada plataforma são definidas e, por fim, é realizada uma transformação automática desse modelo de aplicação para o código da aplicação correspondente à plataforma (modelo de

aplicação) escolhida. O processo garante que há uma equivalência funcional entre toda primitiva definida no esquema conceitual com sua(s) respectiva(s) primitiva(s) no modelo de aplicação. No exemplo da figura 10.7, foi escolhido um modelo de aplicação (plataforma) constituído por uma arquitetura de três camadas: lógica da aplicação, persistência e interface com usuário.

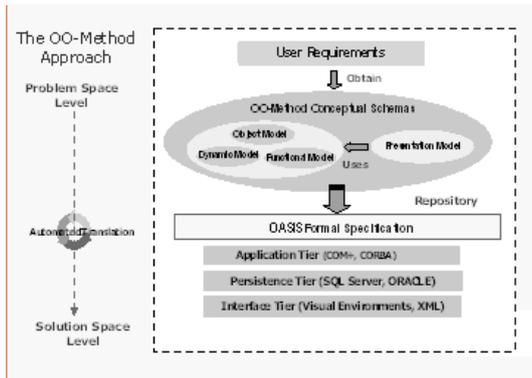


Figura 10.7. Abordagem OO-Method

Comment [A107]: Fonte??

10.3.1.2. Comparação com MDA

Os modelos do OO-Method, seus mapeamentos e transformações podem ser comparados com o padrão MDA, através da tabela [tabela abaixo](#):

Tabela 10.1. Comparação do OO-Method com MDA

MDA	OO-Method
Platform-Independent Model (PIM)	Conceptual Model
Platform-Specific Model (PSM)	Application Model
Implementation Model (IM)	Application Code
PIM-to-PSM transformation	Mappings
PSM-to-IM transformation	Transformations

Comment [A108]: Fonte?? Sugiro adaptar a tabela e colocar em português

Entretanto, algumas propriedades do OO-Method estão ausentes no padrão MDA (OMG), tais como:

- O processo de compilação de modelo
- Em termos de PIM, OO-Method provê uma solução semanticamente precisa, porque está especificado usando uma linguagem formal OASIS que é computacionalmente completa. Em outras palavras, os modelos do esquema conceitual do OO-Method são também computacionalmente completos e contém toda a informação que é necessária para gerar automaticamente código-fonte.

Essas propriedades não são vistas em abordagens que usam UML (Padrão MDA).

10.3.1.3. O Modelo Conceitual

O modelo ou esquema conceitual composto por quatro visões ou modelos que representam os requisitos funcionais de uma aplicação. Esses modelos são: modelo objeto, modelo dinâmico, modelo funcional e modelo de apresentação.

O Modelo Conceitual do OO-Method primou em ter uma notação visual (gráfica) parecida com UML, mas que só usa apenas parte dos seus conceitos (diagramas) que julga necessário e suficiente para representar um sistema de informação. Além disso, a grande diferença, quando comparado com UML, é que o modelo conceitual do OO-Method tem uma semântica e sintaxe bem precisas [sintaxe bem precisa](#) e, como base, a linguagem formal OASIS. Isso propicia a validação automática do modelo conceitual a fim de não deixar passar falhas [passarem falhas](#) (erros) para os modelos posteriores de mais baixos níveis.

Nas próximas seções, serão apresentadas, de modo geral, as principais características desses modelos. Para detalhes mais específicos, consultar referência bibliográfica [Pastor and Molina 2007].

- **Modelo Objeto**

O modelo objeto especifica as propriedades estáticas do sistema, definido pelo diagrama de configuração de Classe que é composto por:

- Classes
 - Atributos
 - Precondições e Serviços
 - Restrições de Integridade
- Relacionamento entre as Classes
 - Associação, Agregação e Composição [Composição](#).
 - Herança
- Agentes

OO-Method também representa precisamente os tipos relacionamentos como associação, agregação e composição. A associação considera aspectos de cardinalidade, papéis (roles) e também de temporalidade. A temporalidade de uma associação se define como estática ou dinâmica, conforme a associação seja constante (estática), desde o momento em que é criada até o fim do seu tempo de vida.

Além da precisão que o OO-Method trata os conceitos de associação, agregação e composição, ele também considera quais são os impactos que eventos de inserção, deleção e mudança de objetos têm sobre esses relacionamentos entre as classes.

O OO-Method suporta também os conceitos de Herança (generalização e especialização) e herança múltipla. Por fim, o modelo conceitual lida com gerenciamento de complexidade do modelo através da definição de subsistema que tem uma notação visual igual a de uma package em UML.

- **Modelo Dinâmico**

Este modelo representa o comportamento do sistema, especificando suas propriedades dinâmicas através de dois diagramas:

- Diagrama de Transição de Estado
Especifica o ciclo de vida válido dos objetos de uma classe e seus serviços disponíveis em cada estado.
- Diagrama de Interação de Objeto

Especifica as interações válidas entre os objetos através das transações, operações e gatilhos.

Um gatilho é uma condição sobre um estado do objeto que, tornando-se verdadeira, faz com que este objeto dispare eventos ou transações sobre si mesmo (self) ou sobre outros objetos (object) do sistema. A sintaxe de gatilhos na linguagem formal OASIS é: **<destination>::<condition>:<service>**

Onde, **<destination> ::= self | object | class | for all**

Sendo que “self” significa para a si mesmo, “object” para uma instância de outra classe, “class” para todas as instâncias da classe e “for all” para um subconjunto de objetos de uma classe.

- **Modelo Funcional**

O modelo funcional especifica o relacionamento estático e dinâmico através de:

- Definição semântica relacionada às transições de estado
- Descrição de como a execução dos eventos muda o valor dos atributos das classes

O modelo funcional trata também questões de eventos de criação e destruição de objetos, além de transações e operações que afetam os estados (valores dos atributos) dos objetos. O modelo funcional do OO-Method, combinado com sua linguagem de fórmula, provê de modo completo e preciso uma solução para especificar os aspectos funcionais de um sistema via modelo conceitual.

O recurso de “Action Semantics” da UML 2.0, para suprir essa necessidade de modelagem de aspectos funcionais, não possui uma semântica definida claramente e precisamente, como também não os têm os elementos da UML. Assim, o modelo funcional do OO-Method e sua linguagem de fórmula, proveem solução adequada, permitindo:

- Acesso a dados de acordo com o Modelo Objeto
- Definição de lógica seqüencial
- Manipulação de classes e objetos
- Manipulação de relacionamentos
- Uso de operadores lógicos, aritméticos e relacionais

- **Modelo de Apresentação**

O modelo da apresentação especifica os requisitos de Interface de Usuário, modelando uma interface abstrata que é independente de plataforma ou dispositivo. Esse modelo de apresentação a nível de em nível de análise (conceitual) é considerado uma inovação do OO-Method, em relação outras abordagens que, na maioria, descrevem interface-usuário apenas em nível de implementação. No OO-Method, o modelo de apresentação é organizado em três níveis:

- **Nível 3:** Nível mais baixo, constituído pelos elementos básicos de entrada de dados, seleções, grupos de dados, filtros, critérios de classificação, conjunto de visualização, ações e navegação.
- **Nível 2: Unidades de Interação**
Nível intermediário com conceito fundamental do modelo de apresentação que descreve um particular cenário de interação entre o usuário e o sistema. Geralmente, a interface de usuário de um sistema é definida como uma coleção relevantes unidades de interação e pelo modo como essas unidades estão estruturadas. OO-Method provê quatro tipos de unidades de interação, descritas a seguir:
- **Nível 1: Árvore de Hierarquia de Ação**
Nível mais alto. Uma vez definidos os cenários de interação do nível 2 do modelo de apresentação, faz-se necessário determinar como essas unidades de interação serão estruturadas e apresentadas ao usuário. Essa estrutura caracteriza o nível mais alto da interface com o usuário, o que poderia ser descrito como o “Menu” principal da aplicação. A árvore de hierarquia de ação serve para esse propósito. Ela é estruturada hierarquicamente por uma árvore, tendo um nó raiz e respectivas ramificações até chegar às folhas. Por exemplo, um sistema de aluguel de carros (nó raiz) é organizado principalmente como tendo as seguintes ramificações: Veículos, Clientes, Aluguéis e Usuários.

A construção do modelo de apresentação pode ser realizada de modo “top-down”, ou seja, partindo-se do nível 1 até chegar aos elementos do nível 3; ou “bottom-up”, partindo-se do nível 3 até a definição do nível 1.

Além dessas quatro visões do modelo conceitual, o OO-Method dá suporte a interoperabilidade e interface do sistema modelado com outros sistemas externos existentes, chamados de sistemas legados, através do conceito de visão de legado.

Enfim, ao se definir os quadro modelos básicos (objeto, dinâmico, funcional e apresentação) do esquema conceitual do OO-Method, tem-se toda infra-estrutura necessária e suficiente para representar um sistema de informação no contexto do espaço do problema.

10.3.1.4. O Compilador de Modelos

Como OO-Method segue o processo ideal MDD de transformação de modelos, ele precisa de alguma forma transformar o modelo conceitual, que contém todas as propriedades estáticas, dinâmicas e de interação com usuário (apresentação), em um modelo de implementação (código). Essa transformação é automatizada através do Compilador de Esquema Conceitual que pode ser visto como uma máquina virtual de programação. OO-Method inova com essa idéia de compilador de modelos, diferindo-o de outras abordagens que apenas focam a geração de código a partir de modelos através de técnicas diversas como integração, sincronização de código, etc. [etc.](#)

Em comparação com outras abordagens existentes, o OO-Method se destaca por tratar de modo completo e preciso todos os aspectos da compilação de modelo. Um exemplo típico de problemas com outras abordagens são as insuficientes transformações para se obter o produto final de software. Mais grave ainda, a maioria das ferramentas que geram código a partir de modelos, consideram [grave ainda, a maioria das ferramentas que geram código a partir de modelos, considera](#) única e exclusivamente como seu modelo inicial de entrada o diagrama de classes em UML, negligenciando todos os demais diagramas que capturam os demais aspectos dinâmicos e de interação de uma aplicação.

10.3.1.5. OLIVANOVA

O OO-Method é implementado através do produto OlivaNova da Care Technologies [OlivaNova 2009]. O principal objetivo de OlivaNova é separar o que deve ser feito (espaço do problema), de como deve ser feito (espaço da solução). Ela é composta de duas principais ferramentas: o modelador e a máquina de transformação. O modelador permite:

- Modelar objetos e negócios;
- Modelar dados;
- Modelar integração;
- Modelar sistemas legados;
- Modelar regras e limitações;
- Definir conceitualmente interfaces do usuário;
- Suporte a UML;
- Suporte a XML.

A máquina de transformação é que implementa todo o processo de compilação de modelos do OO-Method, conforme descrito na seção anterior 10.3.1., gerando código fonte na plataforma de destino.

No sítio da OlivaNova [OlivaNova 2009] existe um quadro que a compara com várias outras ferramentas MDD comerciais, tais como Borland Together, Compuware OptimalJ, IBM Rational Software Architect, etc.

10.3.2. AndroMDA

AndroMDA [AndroMDA 2009] é uma poderosa ferramenta MDA Open Source. Possui arquiteturas como Spring, EJB, .Net, Hibernate, Struts e está desenvolvida sobre o Eclipse. Pode ser utilizada pelos servidores de aplicação Jboss e TomCat e suporta a UML2.0. Agora está em sua versão 4.0, disponível somente para “preview” e permite a criação e utilização de metamodelos no padrão EMF (Eclipse Model Framework) [Projetos Eclipse 2009] e, além disto, possibilita a definição de transformação de modelos PIM a modelos PSM para depois atingir a geração de código fonte, fazendo uso de transformações Model to Text.

Como framework, gerencia qualquer tipo de modelo (geralmente modelos UML guardados no formato XMI) produzido por outras ferramentas case, que combinados aos plugins que possui, permitem [permite](#) a geração de modelos e código fonte.

Em AndroMDA é possível gerar componentes para todas as linguagens: Java, .Net, HTML, PHP. Se desejarmos utilizar alguma tecnologia que ainda não esteja contemplada, somente temos que desenvolver plugins para isto (ou mudar algum que já exista). É mais comumente utilizado por programadores da tecnologia J2EE, inclusive

podendo gerar um projeto J2EE e seu código partindo de um modelo de classe. É possível definir gerar código para Hibernate, EJB, Spring, WebServices e Struts e o código gerado é automaticamente adicionado ao projeto e ao processo de compilação.

Sua realidade MDA permite fazer com que o trabalho de arquitetura e desenvolvimento seja mais curto e de mais qualidade, trabalhando com modelos independentes de plataforma que posteriormente serão refletidos em modelos UML (PSM). Isto permite, entre outras vantagens, ter foco no modelo e necessidades organizacionais (Modelo PIM) e a possibilidade de reutilizar o modelo PIM em outros projetos.

Como etapa seguinte se pode efetuar a transformação até o código fonte da aplicação, tendo como etapas intermediárias a geração de um ou mais modelos PSM. Neste ponto, é onde AndroMDA mais se destaca, por possuir muitos plugins já desenvolvidos e que realizam a transformação PIM > PSM em muitos tipos de linguagens e tecnologias diferentes. Estes plugins são chamados cartuchos "cartridges" e utilizá-los são bastante fáceis.

Além das vantagens citadas anteriormente, destacamos como pontos positivos de AndroMDA: não desenvolvimento de código redundante, o código reflete exatamente o que definem os modelos e a possibilidade de alterar de "cartridge" para que gere o mesmo sistema em outras linguagens. Para se desenvolver novos cartuchos para qualquer plataforma específica deve-se basicamente identificar as regras de transformação e criar um perfil (profile) UML

Entretanto, o nível mais alto de abstração de AndroMDA depende da ferramenta de modelagem que gera UML (PIM). Assim, o nível mais alto é o conceito de caso de uso. A ferramenta de melhor aceitação para modelar em UML e, fazer exportação do metamodelo UML em XMI é a Magicdraw.

AndroMDA não provê recursos de definição de interface usuário abstrata tal como existe em OO-Method. AndroMDA usa conceito de sincronização de modelos (PIM E PSM) e trata questões de rastreabilidade e validação de modelos de forma limitada.

A versão AndroMDA 4.0 que está sendo desenvolvida visa aperfeiçoar o processo de transformação de modelos e, principalmente, receber como entrada "input", no modelo inicial de mais alto nível, metamodelos de qualquer linguagem de domínio específico. Isso, tornará o ambiente MDD de AndroMDA capaz de importar qualquer outro modelo de sistema, e não apenas, UML.

10.4 Problemas e Desafios dos Processos MDD

10.4.1. Visão Geral

Pode-se considerar que desenvolvimento de software dirigido a modelos ainda não está num nível de maturidade suficiente para realizar o sonho acalentado de todo desenvolvedor: ter um produto final de software com qualidade e 100% gerado automaticamente a partir dos seus requisitos. Nos melhores ambientes, o desenvolvedor ainda consegue modelar a nível de [em nível de](#) análise e projeto, mas não a nível de requisitos. Com isso, o desenvolvedor, dessa primeira década do terceiro milênio, ainda realiza esforço manual considerável a nível de [em nível de](#) projeto e implementação. E isso, tem impactos negativos nos custos, prazos e qualidade dos softwares produzidos. O paradigma dirigido a modelos traz uma promessa de tornar realidade esse sonho. Entretanto, muitos desafios haverão de ser superados.

Sabe-se que processo MDD ainda está na sua infância. Nem as linguagens (modelos) nem as ferramentas se desenvolveram o suficiente para concretizar suas promessas feitas. O processo MDA, padronizado pela OMG, é apenas uma referência e pode ser usado por qualquer outro processo específico de desenvolvimento de software existente (RUP, XP, OPEN, Agentes, Aspectos, Formais, etc.) desde que se adapte e seja dado um foco especial em modelos e suas transformações [dados um foco especial em modelos e suas transformações](#). Decerto que processos como RUP e OPEN, por suas próprias características, são mais fáceis de serem adaptáveis a um processo dirigido a modelos do que o XP cujo foco predominante é implementação.

Este capítulo do livro procurou relatar alguns aspectos dessa tecnologia MDD. Como trabalhos futuros e desafios para o processo desenvolvimento de software dirigido por modelos, destacam-se os seguintes:

- Integração com a fase requisitos (Engenharia de Requisitos) para elevação do nível de abstração inicial a ser modelado (modelo CIM da MDA);
- Suporte a modelos orientados a metas “goals”, agentes e aspectos.
- Melhor precisão semântica dos modelos em relação às características estáticas, dinâmicas e de apresentação (interação-usuário);
- Melhores mapeamentos entre os modelos;
- Melhor transformação automática de modelos (automação);
- Melhor suporte à Validação de Modelos;
- Melhor integração com as plataformas específicas (PSM);
- Melhor e maior percentagem de código fonte gerado;
- Maior suporte à rastreabilidade;
- Melhor suporte à engenharia reversa;
- Suporte à computação autonômica;
- Suporte a ontologias.

10.4.2. Lições Aprendidas na adoção de soluções MDA

Muitas organizações que, nos últimos anos, vêm utilizando com sucesso soluções MDA, perceberam que um conjunto de práticas e passos consistentes devem ser considerados ao se adotar um processo MDD automatizado. A seguir, é mostrado um resumo com os passos necessários [mostrados um resumo com os passos necessários](#) para se desenvolver uma solução MDA adequada:

- Examinar os modelos atualmente usados na empresa no seu processo de desenvolvimento e a conexão/correlação semântica entre os elementos desses modelos.
- Identificar as transformações candidatas para automação
- Especificar (documentar) os requisitos dessas transformações
- Criar os UML Profiles necessários
- Desenvolver o código da(s) transformação(coes)
- Esboçar documentos de uso, empacotar e distribuir [distribuir](#).

10.4.3. O programa FastStart da OMG

Recentemente, o grupo OMG lançou o programa “FastStart” para ajudar as organizações aprenderem sobre MDA e aplicar MDA nas arquiteturas de seus sistemas, na integração dos sistemas e nos seus processos de desenvolvimento de software. Durante programa FastStart, a organização recebe consultores da OMG para realizarem as seguintes atividades:

1. Análise inicial MDA
2. Revisão da Arquitetura Empresarial MDA
3. Plano de Transição MDA
4. Seminários Executivos MDA
5. Prática MDA

Essas atividades duram em média 5 semanas e ajudam à equipe executiva e técnica da empresa a:

- Analisar claramente e planejar como MDA pode melhor ser introduzido e aplicado para melhor beneficiar a organização e seus processos de negócios chaves
- Capacitar a empresa em MDA para que ela própria, sem ajuda de provedores externos, desenvolva seu processo MDA/MDD.

10.5. Tópicos de Pesquisa

O Processo de desenvolvimento de software dirigido por modelos (MDD) ainda é um tema recente. Os benefícios do padrão MDA ainda não foram bem entendidos pelas organizações. Existem várias linhas e tópicos de pesquisa relacionados com MDD/MDA, entre estes se destacam:

1. Desenvolvimento de modelos precisos semanticamente e completos para facilitar transformações e validações.
2. Desenvolvimento ou aperfeiçoamento de ferramentas de apoio ao processo MDD
3. Adaptações do processo MDD aos processos específicos de desenvolvimento de software
4. Extensões MDA para novas plataformas
5. MDA em Linhas de Produtos de Software(LPS)
6. Rastreabilidade em processos MDD
7. Medição/Estimativas de projetos de software em ambientes MDD
8. MDD para sistemas embutidos (Embedded systems development)

10.6. Sugestões de Leitura

Este capítulo propôs dar uma visão sobre Desenvolvimento de Software Dirigido por Modelos. Entretanto, devido ao enorme escopo e dinamismo dessa área, realizou-se um esforço considerável para contemplar, bem como resumir de modo claro e objetivo, um maior número de tópicos possíveis sobre o tema, ou seja, tentou-se elaborar um trabalho que fosse o mais científico, atualizado e completo possível. Porém, sabe-se que qualquer estudo, por mais minucioso que seja, está longe de esgotar o assunto.

Para complementar o material apresentado neste capítulo, o autor sugere que o leitor realize as seguintes leituras:

- UML executável (OMG): Usando apenas os diagramas de classes, de estado e a extensão UML "Action Semantics" torna um modelo UML capaz de ser executável e o transforma, através de compiladores de modelos específicos, em plataformas como C++, Java, etc.
Para mais detalhes sobre esse tópico, ver livro: **Executable UML, A Foundation for Model Driven Architecture**, Mellor-Balcer, Addison-Wesley, 2002 e o site http://en.wikipedia.org/wiki/Executable_UML.
- Grupo de pesquisa "Precise UML- PUML": Como visto nas seções deste capítulo 10.2.1[France and Ghosh 2006] e 10.3.1 sobre o OO-Method, UML carece de precisão semântica, por exemplo, dependendo da interpretação do projetista pode-se modelar uma um determinado objeto como agregação, composição, associação ou herança. Visando aperfeiçoar a precisão semântica e formalidade da linguagem de modelagem de propósito geral UML, há um grupo de trabalho desenvolvendo Precise UML-PUML em <http://www.cs.york.ac.uk/puml/maindetails.html>.
- Ambiente MDD "Moskitt" da Universidade Politécnica de Valencia: O Kit de Modelagem de Software (Modeling Software KIT-MOSKitt) é um ambiente case MDD de código aberto (livre) construído sobre o Eclipse IDE, desenvolvido pelo Ministério Regional de infra-estrutura e transportes de Valencia (Espanha). Para maiores informações ver <http://www.moskitt.org/eng/moskitt0/>.
- Ambiente MDD Open source "OPENMDX": Ferramenta concorrente do AndroMDA, roda no Eclipse IDE, sendo considerado também o estado da arte em desenvolvimento dirigido a modelos. Boa documentação adicional, inclusive ferramenta disponível para download em <http://www.openmdx.org/>.
- Livros específicos sobre o tema:

- **MDA Explained: The Model Drive Architecture:Pracice e Promise** by Anneke Kleppe, Jos Warmer, Wim Bast (Editora Addison Wesley 2003): Obs. Muito bom para uma introdução sobre MDA.
- **Model-Driven Software Development** by Sami Beydeda, Matthias Book , Volker Gruhn (Editora Springer 2005): Obs. Excelente pelo nível teórico básico , avançado e técnico.
- **Real-Life MDA: Solving Business Problems with Model Driven Architecture (Interactive Technologies)** (Editora Morgan Kaufmann, 2006): Obs. Totalmente prático, focado em soluções de processos de negócios, utilizando-se MDA/MDD.

10.7. Exercícios

Para sedimentar melhor os conhecimentos teóricos abordados nesse capítulo, os seguintes exercícios foram elaborados:

1. MDA usa modelos distintos para separar os sistemas/aplicações em os níveis de abstrações/visões distintos. Quais são modelos definidos no padrão MDA e descreva seus propósitos.
2. Descreva a técnica de transformação de modelos através de metamodelos.
3. Quais os padrões que estão no núcleo da arquitetura MDA do grupo OMG.
4. Discuta: MDA determina o uso ou recomenda algum processo específico de desenvolvimento de software, tal como RUP, XP ou outro qualquer? Como se poderia adaptar o RUP ou XP para utilizar um processo dirigido a Modelos (MDD/MDA)?
5. Caracterize o modelo conceitual do OO-Method.
6. O que é o Compilador de Modelos do OO-Method?
7. Compare as ferramentas CASE de desenvolvimento de software dirigido a modelos: OLIVANOVA e AndroMDA.
8. Suponha uma ferramenta MDD que dá suporte completo aos modelos CIM, PIM e PSM. Durante as manutenções dos aplicativos desenvolvidos com essa ferramenta qual é o único _desse modelos que, segundo o padrão MDA, deve ser alterado?
9. Quais os benefícios em se adotar uma processo [um processo](#) de desenvolvimento de software dirigido a modelos?
10. Usando uma ferramenta de modelagem UML (MagicDraw, ArgoUML,etc) modele a seguinte aplicação simplificada para administração de alunos, utilizando o diagrama de classe da figura 10.8. As funcionalidades (operações) básicas da classe Aluno são inserir, excluir, listar e alterar, conforme diagrama de atividades da figura 10.9. Depois, instale uma das ferramentas free MDD (AndroMDA ou OPENMDX), configure-as adequadamente e importe/utilize o modelo UML. Por fim, tente usar ferramenta MDD para gerar seu aplicativo na plataforma JAVA JEE, inclusive com recursos de persistência num banco escolhido (Mysql, Postgresql, etc) e “deployment” num servidor JSP Tomcat Apache. Talvez, você tenha tido dificuldade em instalar e configurar esses ambientes de desenvolvimento, mas que achou do processo de transformar seu modelo PIM em PSM automaticamente? Que achou de obter o código do aplicativo automaticamente? Tente fazer uma alteração (evolução) do aplicativo para adicionar disciplinas, professores e seus respectivos relacionamentos com alunos, lembrando-se de que a alteração deverá ser realizada apenas no modelo PIM (mais alto nível de abstração).

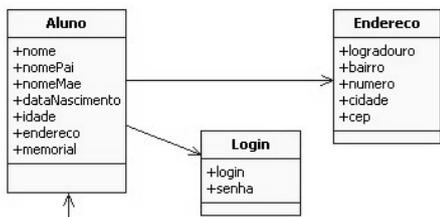


Figura 10.8 Diagrama de Classes

Comment [A109]: Fonte:??

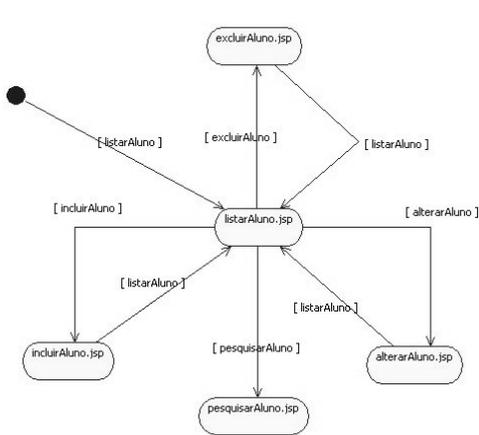


Figura 10.9 Diagrama de Atividades

Comment [A110]: Fonte:??

Referências

- MDA productivity case study. The Middleware Company (2003). <<http://www.omg.org/mda/presentations>>. Acesso em julho 2009.
- OMG: padrão MDA (2003).< <http://www.omg.org/mda> >. Acesso em julho 2009.
- Pastor O.; Molina J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer Publisher 2007.
- OlivaNova Care Technologies, Denia, Spain (<<http://www.care-t.com>> Acesso em julho 2009.
- Kontonya, G. e Sommerville, I. (1998) Requirement Engineering: Processes and Techniques, John Wiley & Sons.
- MDA Guide Version 1.0.1, Document Number: omg/2003-06-01 Date: 12th June 2003. <http://www.omg.org/mda/>
- Hitachi M. O.; MDA and System Design. Presentation at MDA Information Day, OMG Technical Meeting, April 2002.
- Model Driven Architecture (MDA). Document number ormsc/2001-07-01, Architecture Board ORMSC1, July 9, 2001. Disponível em <<http://www.omg.org/mda/presentations>>. Acesso em julho 2009.
- France R. B.; Ghosh S.; Dinh-Trong T. : Model-Driven Development Using UML 2.0: Promises and Pitfalls. In IEEE *Computer*, vol. 39, no. 2, pp. 59-66, Feb. 2006.
- Morgan T (2002) Business rules and information systems – aligning IT with business goals.Addison-Wesley, Reading,MS.
- Pastor, O.: Model-Driven Development: The OO-Method Approach. Presentation at UFPE, Recife, Brasil August 2008.
- AndroMDA. <<http://www.andromda.org>>. Acessado em julho 2009.
- Projetos Eclipse <<http://www.eclipse.org/projects/>> Acesso em julho 2009.

Capítulo 16

Modelagem de Processos

André Luis Rodovalho Bezerra

1. Introdução

Modelagem de Processos significa desenvolver diagramas (Diagramas de Processos) que mostram as atividades da empresa, ou de uma área de negócios, e a seqüência na qual são executadas. Muitos negócios são relativamente complexos, assim um modelo poderá consistir de diversos diagramas, e o alvo da modelagem é ilustrar um processo completo, permitindo aos gestores, consultores e colaboradores melhorarem o fluxo e aperfeiçoarem o processo.

É para a construção desse diagramas atualmente temos algumas linguagens para a construções desse diagramas de processos, onde neste capítulo iremos focar nas linguagens **BPMN e SPEM** alguns dos seus objetivos, especificações e notações, mostrando algumas ferramentas existentes para modelagem das mesmas e finalizando com uma comparação entre elas.

1.1. O que é Modelagem de Processos

Modelar processos ajuda a entender como funciona uma organização. Modelar um processo pode ser bastante difícil na prática, principalmente quando é a primeira vez, e lembrando que um processo pode permear diversas áreas funcionais, o que requer um trabalho conjunto de pessoas destas áreas funcionais. Durante este trabalho, os participantes apresentam um aumento do entendimento do negócio.

O **modelo** é um ponto central para que os participantes definam mudanças para melhoramento do processo ou mesmo um desenho completamente novo. Pode ser identificado se um processo é eficiente e **eficaz**, ou mesmo antecipar sua complexidade, redundâncias e não conformidades (problemas). Se o processo é alguma coisa nova que

Comment [A111]: Formatar Tabulação para 1,27cm (ver os demais parágrafos)
- Também na formatação de todos os parágrafos, o espaçamento antes é de 6 pontos

BRANÇO

Comment [A112]: Não devem existir espaços (linhas em branco) entre as seções e/ou parágrafos, basta ajustar a formatação do parágrafo para ter espaçamento antes de 6 pontos

Comment [A113]: Evitar colocar / . O mais correto seria usar o sinal de pontuação parenteres: (eficaz)

A **comunicação do processo**, de forma eficiente, para outras pessoas é fundamental. Por melhor que seja um processo, se a comunicação para outros for deficiente, principalmente para aqueles que vão implementar o processo, o esforço desenvolvido pela equipe terá sido em vão. Bons modelos de processos ,são a chave para a comunicação.

Os resultados da modelagem de um processo são essencialmente: acréscimo de valor para o cliente e redução de custos para a empresa. O que, conseqüentemente, conduz a empresa ao aumento de lucros. Um diagrama de modelo de processo de negócio é uma ferramenta, ou seja, um meio para se atingir um fim determinado e não um resultado de desempenho por si

só.

A saída final deve ser a melhoria na maneira como o processo do negócio funciona, o foco das melhorias está nas ações que geram valor agregado ao negócio, ou seja, que melhoram o serviço e a experiência do cliente, além de reduzir tempo e esforço gastos. Utilizamos BPI(Business Process Improvement), para a melhoria de processo de negócio, ele é uma metodologia (abordagem) que ajuda a otimizar e entender os processos de negócio com objetivo de alcançar as metas e melhorar os resultados dos processos.O primeiro passo da BPI é determinar o cenário atual dos processos, ou seja, AS-IS(que o cenário atual) e depois definir o cenário futuro, ou seja, TO-BE.

Comment [LA114]: Reveja o espaçamento da próxima linha, lembrando que somente o primeiro paragrafo nao existe tabulação, os demais segundo o template deve haver um recuo de 1,27 cm.

OS dois tipos principais de modelos de processo de negócio:

- Modelo **“as is”** or baseline (a situação atual);
- Modelo **“to be”** (a nova situação pretendida).

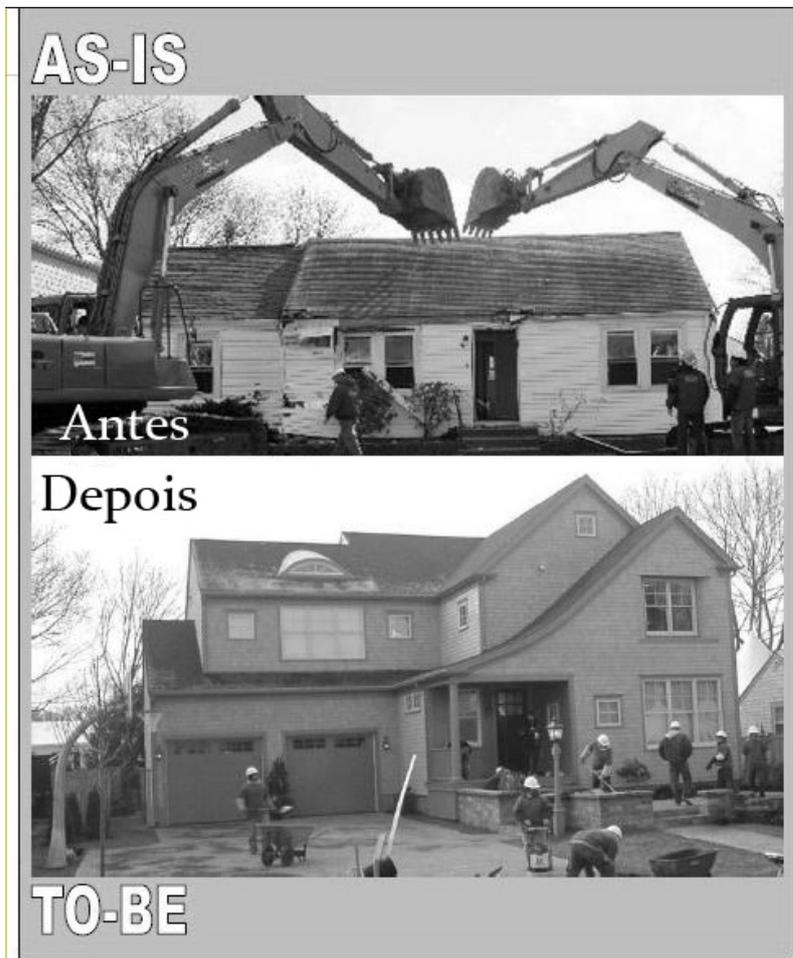


Figura x- exemplo de define os modelos AS-IS e TO-BE

Os quais são utilizados para analisar, testar, implementar e melhorar o processo. outras conseqüências secundárias que se alcançam com a modelagem de processo bem sucedida podem ser o aumento da vantagem competitiva, o crescimento no mercado, e a melhoria de moral e retenção de colaboradores.

Não há nenhuma regra absoluta para o escopo ou extensão de um modelo de processo Em termos de departamentos e atividades cobertas.

Comment [LA115]: Essa imagem, gostaria de ver o real sentido dela, acho que poderia ficar melhor num slide, agora num livro você poderia rever.

Comment [LA116]: tabulação

Comment [LA117]: reveja o espaçamento ao final da frase

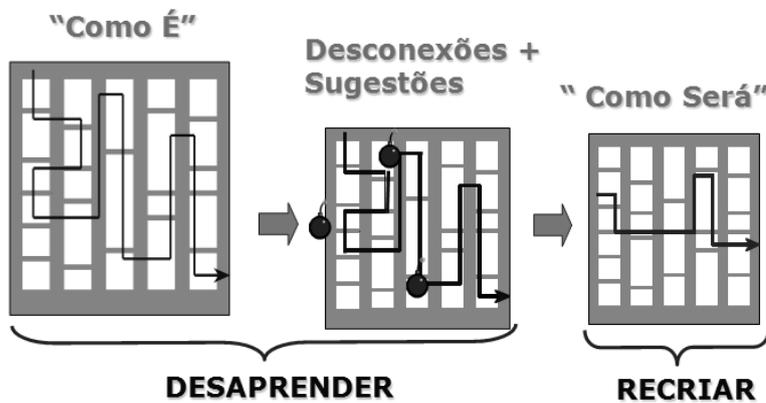


Figura 1 (Modelagem de Processos)

2.1. Objetivo da Modelagem de Processos Software

A modelagem de processos de software tem sido utilizada na engenharia de software ao longo dos anos para melhor entender, gerenciar e controlar o processo de desenvolvimento. Contudo, a descrição dos processos do cliente oferece uma nova perspectiva aos engenheiros de software: a necessidade de diferentes abordagens e o uso de diferentes notações e técnicas.

O principal objetivo da modelagem de processos é representar os processos de uma maneira clara e formal em diferentes níveis de abstração. A disponibilidade de modelos completos permite uma análise crítica das atividades existentes para definir melhorias e racionalizações dos processos.

3.1 Vantagens e Desvantagens da Utilização de Modelagem de Processos

A. Vantagens

Temos algumas vantagens com a modelagem de Processos:

Comment [A118]: Não tenho certeza, mas acho que as figuras devem ter fontes de onde foram extraídas (referências bibliográficas)
- Fonte de figuras deve ser Helvética 10

Comment [LA119]: Deve acrescentar a preposição de entre processos e software.

Comment [A120]: Suponho que esta deva ser a segunda seção do capítulo, ou seja, 16.2 . Portanto, deveria ser com fonte tamanho 13

Comment [LA121]: Primeiro paragrafo nao deve haver recuo.

Comment [A122]: Fonte 13, seção 16.3

Comment [A123]: Não precisa colocar Vantagens e Desvantagens como subseções, da seção 16.3, basta colocá-as em destaque ou como tópicos num nível superior sem numeração.:

- Ex:
- Vantagens
 - X, Y,Z
 - Desvantagens
 - W,V, T ..

- Bons modelos de processos, são a chave para a boa comunicação.
- Se o processo, é alguma coisa nova que a empresa está planejando executar, o modelo pode ajudar a assegurar sua eficiência desde o início.
- Revelar anomalias, inconsistências, ineficiências e oportunidades de melhoria, permitindo à organização que se compreenda melhor e auxiliando na reengenharia desses processos.
- Fornecer visão clara e uniformizada das atividades, suas razões e formas de execução.
- Utilizar o modelo como um meio para distribuição de conhecimento dentro da organização e treinar as pessoas, ajudando-as a conhecer melhor seus papéis e as tarefas que executam.

Comment [LA124]: Verifique o excess de espaços entre as frases.

B. Desvantagens

4.1. Linguagem de Modelos de Processos

Comment [A125]: Fonte 13, seção 16.4 . Fazer um parágrafo simples de apresentação da seção, antes de falar da linguagem.

4.1.1. BPM

Comment [a126]: Esta seria a seção 16.4.1 do capítulo.

O que é Gestão de Processos de Negócios (BPM) do acrônimo em inglês "Business Process Management", segundo esses autores:

"Um conjunto de técnicas para garantir que os processos sejam continuamente monitorados e melhorados." [RUMMLER, G.; BRACHE]

Comment [A127]: Referência incorreta, falta o ANO de publicação . Idem para as demais do Capítulo

"Desenvolvimento e manutenção de uma arquitetura de processos de negócios, que se utiliza de técnicas, metodologias e gerenciamento humano para garantir que os processos sejam continuamente melhorados e monitorados." [Pinto Filho, J. B.]

BPM , envolve modelagem, execução, monitoramento e análise de processos de negócios e, nesse contexto , "Modelagem de processos de negócios, é o conjunto de conceitos e técnicas que visam a criação de um modelo com os processos de negócio existentes em uma organização. Esta "modelagem" é utilizada no contexto da gestão de processos de negócio. [1]

Comment [A128]: Referência não está conforme padrão SBC. Há outras referências erradas como essa ao longo do capítulo, corrigir !

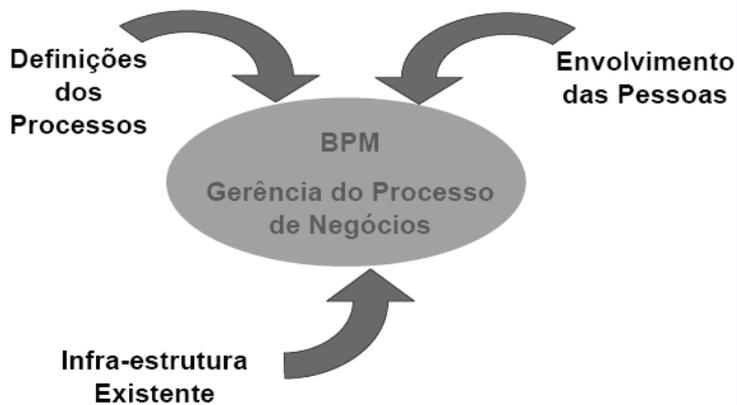


Figura 2 (BPM)

Comment [A129]: - Fonte de figuras deve ser Helvética 10

Dada a similaridade das suas composições, "Funções de Negócio" e "Processos de Negócio" são conceitos que frequentemente suscitam dúvidas entre as pessoas interessadas em formar um melhor entendimento a respeito dos elementos de uma Arquitetura de Negócios. Ambos são "coisas que a empresa faz", entretanto, os processos são transfuncionais (ou horizontais), já que perpassam diversas barreiras funcionais dentro da organização (ex.: adquirir bem, alienar bem, contratar funcionário), enquanto que as funções, que em conjunto descrevem a missão da empresa, são verticais (ex.: contabilidade, vendas, logística).[1]

BPM traz inúmeros ganhos a uma organização, porém não é um simples conceito nem é fácil de implantar, Envolve mudança em estruturas, culturas, processos, para a qual nem todas as organizações estão preparadas.

A tecnologia [figura 3] contribui para o sucesso de um projeto de BPM, mas não é o foco, em primeiro lugar deve-se conhecer e identificar oportunidades nos processos.

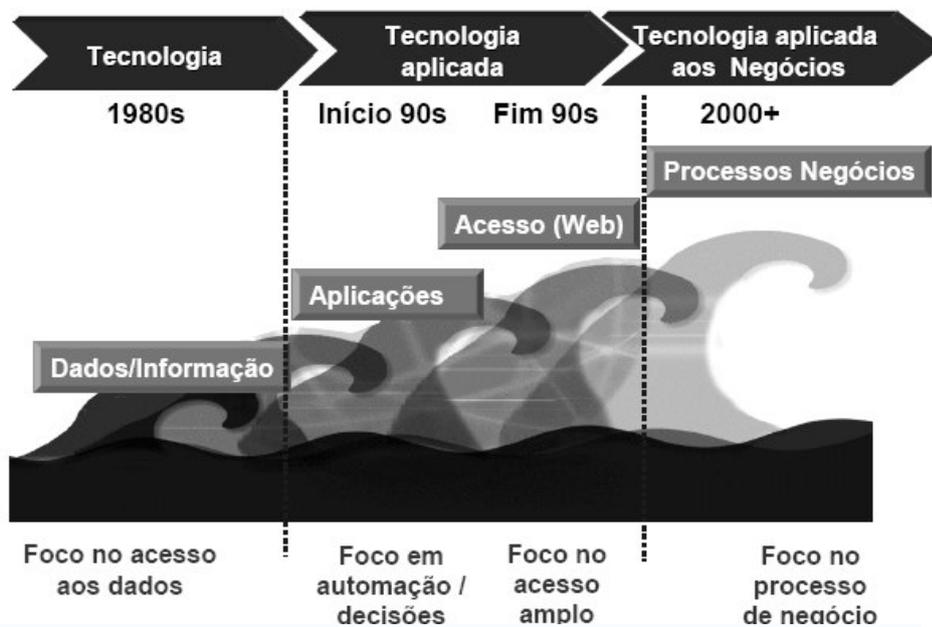


Figura 3 (Evolução de TI “Fonte: Marcos Borges, CHORD – UFRJ, 2003”)

4.1.1.1. BPMN

O Business Process Management Initiative (BPMI) desenvolveu um padrão Business Process Modeling Notation (BPMN). O BPMN especificação 1.0 foi liberado ao público em maio de 2004. Esta especificação representa mais de dois anos de esforços por parte do BPMI Notation Working Group. Em 2005 BPMI se funde com o grupo OMG, e BPMN agora está na versão 1.2 e com uma proposta de especificação da 2.0.

O principal objetivo do esforço BPMN era fornecer uma notação que é facilmente compreensível por todos usuários de negócios, desde os analistas de negócio que criam os rascunhos iniciais dos processos, a desenvolvedores técnicos responsáveis pela aplicação da tecnologia que irá desempenhar essas processos e, finalmente, para as pessoas de negócios que vão gerenciar e monitorar esses processos.[2]

Ela possui uma forma de apresentações mais amigável aos usuários de negócios que os modelos UML tradicionais, permitindo criar uma linguagem comum entre as áreas de negócios e TI, assim permitem aos analistas de negócios entenderem um modelo de processos

Comment [LA130]: Não a necessidade do recuo, somente a partir do Segundo.

Comment [A131]: Falar que em 2005 BPMI se funde com o grupo OMG, e BPMN agora está na versão 1.2 e com uma proposta de especificação da 2.0. Ver <http://www.omg.org/bpmn/>

Comment [LA132]: Reveja o alinhamento, a partir do Segundo paragrafo deve haver um recuo.

processos que sejam entendidos por outros analistas sem a necessidade de treinamento especial.

Segundo definições do BPMN 1.2 o modelo BPMN é dividido em três tipos básicos de sub-modelos:

- privado,
- abstrato
- colaboração.

Os processos de negócio **privados**, mais *baixo nível*, correspondem aqueles que ocorrem dentro da organização e que possuem atividades realizadas internamente que interagem entre si. É utilizado quando se quer visualizar uma parte de um processo sem se preocupar com o processo como um todo.

É o tipo de processo mais comum, composto por uma série de atividades que são realizadas unicamente dentro de uma empresa. O fluxo da sequência do processo é contido dentro do Pool e não pode cruzar os limites do Pool como mostra a figura abaixo.

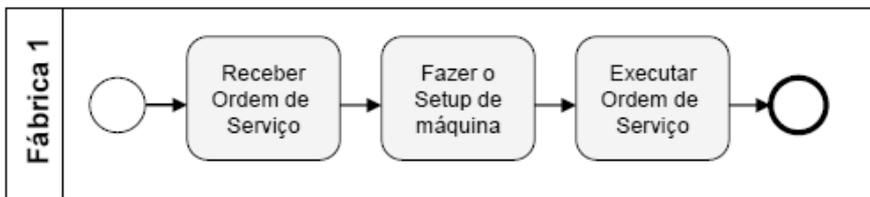


Figura x- Processo de negocio **PRIVADOS**

Processos de negócio **abstratos**, mais alto nível, é utilizado quando se quer visualizar as interações entre os fluxos, as comunicações. São processos públicos que retratam as interações das atividades pertencentes a um processo privado com outra entidade de negócio externa ao processo privado.

Muitas vezes, o processo inclui atividades que são realizadas fora da empresa (realizado por terceiros, por exemplo) e não temos gerência sobre a execução destas atividades. Utilizamos um modelo abstrato para representar uma “entidade” independente, com processos próprios, mas que não podemos modelar (por não conhecer o processo) ou não nos interessa modelá-lo;

No exemplo abaixo o Fornecedor faz o beneficiamento da matéria prima, entretanto, é um processo interno do fornecedor, o qual não é conhecido, ele deve ser modelado como um processo abstrato (caixa preta).

Comment [LA133]: Ponto seguida, deve haver um espaço para iniciar a nova frase.

Comment [LA134]: Recuo.

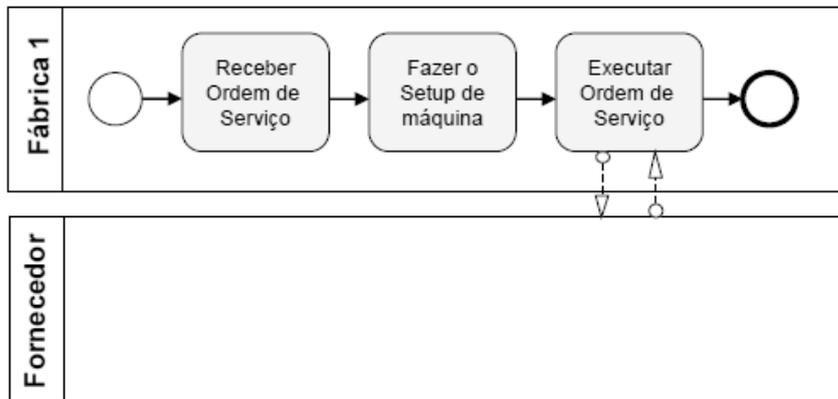


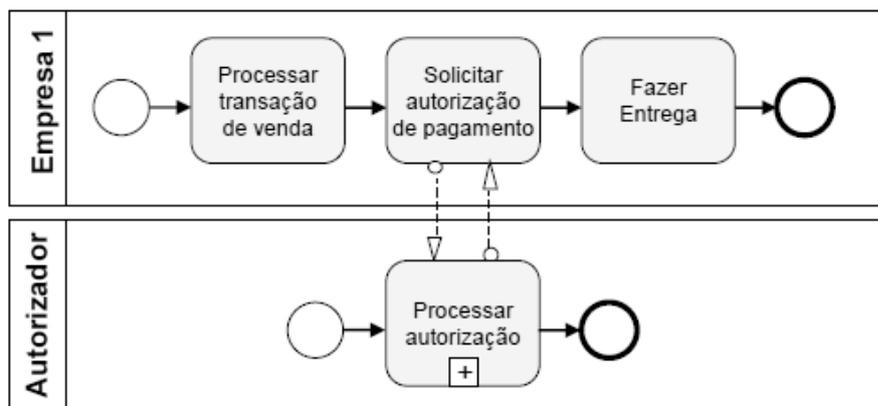
Figura x- Processo de negocio **ABSTRATOS**

Processos **colaborativos** possuem uma visão em alto nível e baixo nível. Este tipo de processo modela as interações entre dois ou mais processos de **negócio**.

Descrevem processos B2B e as interações entre duas ou mais entidades de negócio. Os diagramas processos são geralmente de um ponto de vista global. As interações são descritas como as seqüências de atividades e as trocas de mensagens entre os participantes.

No exemplo abaixo o Autorizador (Administradora de Cartão de Crédito) faz a autorização de pagamento por cartão de crédito, neste caso este processo interessa a Empresa 1 (que realiza a venda), logo ele deverá ser modelado (desenhado) explicitamente.

Comment [LA135]: Ponto final



Em sub modelos são criados BPD (*Business Process Diagram*), Elementos de 4 categorias São eles: objetos de fluxo, objetos de conexão, artefatos e swimlanes. como mostra na figura 4:

Comment [LA136]: Recuo.

<u>1-Objetivos de Fluxos</u>	<u>2-Objetivos de Conexão</u>	<u>3-Artefatos</u>	<u>4-Swimlanes</u>
Eventos	Fluxo de Seqüência	Objetos de Dados	Pools
Atividades	Fluxo de Mensagem	Grupos	Lanes
Gateways	Associação	Anotação	

Figura 4- Categorias Básicas de Elementos BPMN

Fonte: Flávia Monique Baseada em (BPMN, 2009)

Comment [A137]: - Fonte de figuras deve ser Helvética 10
** Idem para todas Figuras do Texto.

1- *Objetos de fluxo*: definem o comportamento

- Evento, Atividade e Gateway



Figura x- Eventos, Atividades e Gateway

2- *Objetos de conexão*: conectores de objetos de fluxo

- Fluxo seqüencial, Fluxo de mensagem e Associação.



Figura x- Sequence Flow, Message Flow e Association

3- *Artefatos*: informações adicionais sobre os fluxos

- Dados, grupo e anotação

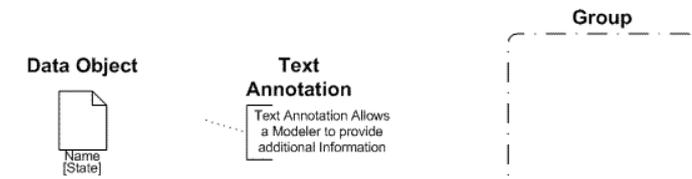


Figura x- Data Object, Text Annotation e Group

- Dividir um diagrama BPMN de acordo com os responsáveis pela execução das atividades;
- Forma fácil de ver onde um fluxo de processo atravessa uma linha funcional ou departamental.

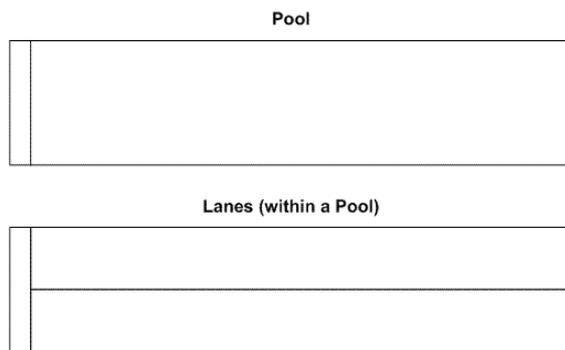


Figura x- POOL e Lanes.

A BPMN define um único tipo de diagrama, chamado de BPD (Business Process Diagram). Como forma de exemplificar isto, observe que na Figura 5.5 são dispostos alguns dos elementos que formam esta notação e o diagrama representa parte da modelagem do processo de compra de um produto em uma loja

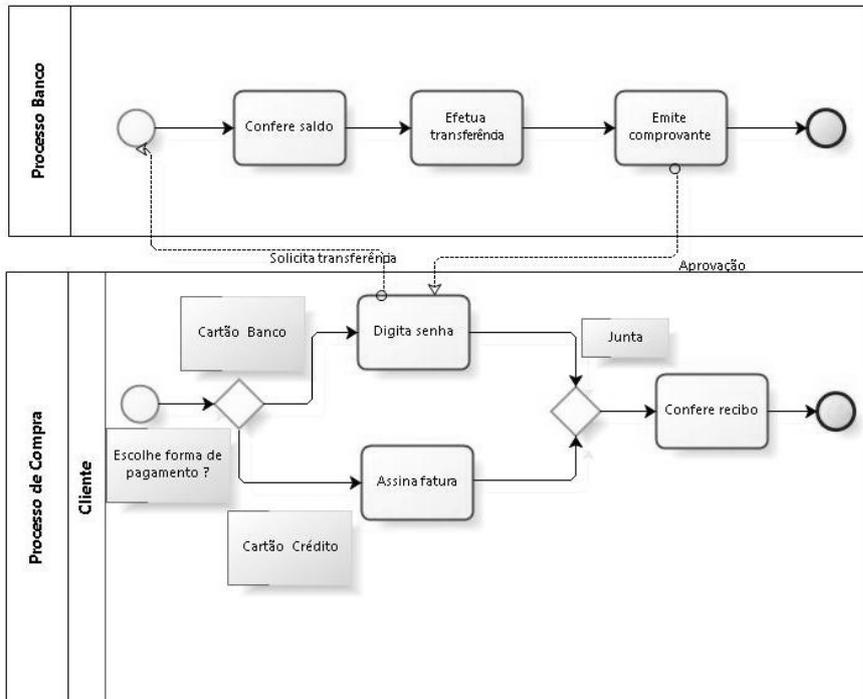


Figura x- Diagrama de Processos de uma loja.

Comment [a138]: Colocar um exemplo de processo modelado com BPMN

4.1.2.SPEM

O SPEM [3], Software Process Engineering Metamodel, é um metamodelo (conjunto de construtores e regras para a criação de modelos) que pode ser usado para descrever um processo concreto ou uma família de processos de desenvolvimento de software relacionados. A execução do processo (enactment) não está no escopo deste modelo.

O SPEM foi desenvolvido e é mantido pelo OMG (Object Management Group) desde 2002 resultado do esforço coletivo de pesquisadores e consultores, tais como: Empresas: IBM, Rational, Computer Associates, Toshiba, Siemens, etc. e pesquisadores como: Philippe Kruntchen, Craig Lairman, e diversos outros. para tentar suprir a necessidade de um padrão para as técnicas de modelagem de processo de software surgidas nos últimos anos. Define o conjunto mínimo de elementos de modelagem necessários para descrever qualquer processo de desenvolvimento de software, utilizando uma abordagem orientada a objetos e a UML (Unified

Modeling Language, linguagem gráfica para modelagem de sistemas discretos, de maior aplicabilidade na área de projeto de software orientado a objetos; a UML 1.4, de Jan 2001, é a versão referenciada pelo SPEM) como notação. É Atualmente o SPEM está na versão 2.0 de

figura 5.

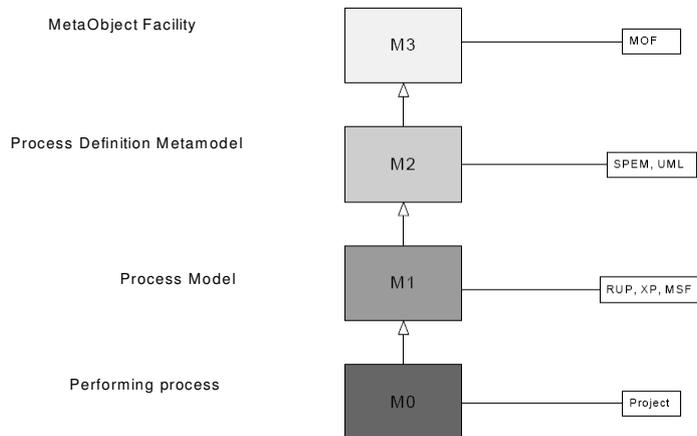


Figura 5- Arquitetura do SPEM definida pela OMG.

O nível M0 é definido por processos instanciados, ou seja, processos que estão em execução em um projeto. A definição dos processos abstratos, estão no nível M1, como por exemplo o Rational Unified Process (RUP). O SPEM se encontra no nível M2 servindo de base para os processos do nível M1. Por fim, no nível M3 está definido o Meta-Object Facility (MOF) que é uma tecnologia adotada pela OMG para definir meta-dados.

O SPEM utiliza mecanismos de extensão da semântica padrão da UML, para adaptá-la ao propósito da modelagem de processos, que são: estereótipos, valores atribuídos e restrições. Um estereótipo, por exemplo, estende o vocabulário da UML, permitindo a criação de novos blocos de construção derivados dos já existentes. Um fator que favorece a escolha do SPEM para a definição de processos é que ele tanto define capacidades de modelagem dedicadas ao domínio do processo de software, quanto se beneficia da expressividade da UML. Assim, desenvolvedores de software que estejam familiarizados com a UML podem reutilizar seus conhecimentos de modelagem de software no domínio da modelagem de processos de software.

4.1.2.1. A notação SPEM

Os diagramas da UML são usados para apresentar os vários aspectos de um modelo de processo de software, como os de classe, de pacote, de atividade, de caso de uso, de seqüência e de estados, com algumas restrições.

Os diagramas de implementação e componentes não são usados porque contêm alguns

Comment [LA139]: Recuo.

entrada e saída, bem como os estados do fluxo de objetos. “Raias” podem ser utilizadas para separar as responsabilidades dos diferentes papéis do processo. Os elementos de definição do processo do SPEM são representados por estereótipos. Ícones especiais foram criados para os mais freqüentemente utilizados, como atividades, produtos de trabalho, papéis, etc (veja alguns exemplos na Tabela 1).

Comment [LA140]: recuo

Tabela 1. Ícones e estereótipos do SPEM.

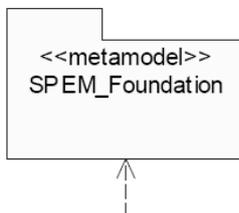
Estereótipo	Comentário	Notação
<i>WorkProduct</i>	Classe de produto de trabalho produzido em um processo e está associado a um tipo de produto.	
<i>WorkDefinition</i>	Descreve o trabalho executado no processo. Suas subclasses são <i>Activity</i> , <i>Phase</i> , <i>Iteration</i> e <i>LifeCycle</i> .	
<i>Activity</i>	Descreve uma parte do trabalho executado por um <i>ProcessRole</i> : as tarefas, operações e ações que são executadas por um papel.	
<i>ProcesRole</i>	Define responsabilidades sobre <i>WorkProducts</i> específicos e é o executor de <i>Activities</i> .	
<i>Document</i>	Diferentes tipos de <i>WorkProduct</i> como por exemplo Documento Texto, um Modelo UML, Executável, Biblioteca de Código, etc.	

AQUI

Comment [a141]: COLOCAR UM EXEMPLO SIMPLES DE PROCESSO MODELADO COM SPEM, pode ser o RUP, o mesmo de um material enviado pelo Prof.

4.1.2.2 Estrutura de Pacotes

O modelo SPEM é estruturado em dois pacotes, como mostrado na figura 6.



SPEM_Foundation → Estende um subconjunto do meta-modelo da UML 1.4. (Data_Types, Core, Actions, State_Machines, Activity_Graphs, Model_management)

SPEM_Extensions → Adiciona as construções e semânticas requeridas para a engenharia de processos de software

Figura 6- Estrutura de Pacotes do SPEM.

O primeiro pacote, chamado *SPEM_Foundation*, representa uma extensão da especificação Unified Modeling Language (UML) e contém os elementos básicos que dão suporte ao SPEM e ao segundo pacote, o *SPEM_Extensions*, que contempla abstrações necessárias e as semânticas requeridas para a engenharia de processo de software [SPEM,2008].

O pacote *SPEM_Foundation* é formado por seis sub-pacotes: *Core*, *Data_Types*, *Activity_Graphs*, *Model_Management*, *Actions* e *State_Machines*. Dentro deles estão contidos elementos básicos para modelagem de processo, como tipos de dados: *Boolean*, *Integer*, *String*, *AggregationKinf*, etc.; elementos que representam ações: *Action*, *CallAction*, etc.; elementos de estados: *State*, *Transiction*, etc.; elementos que representam diagramas de atividades: *ActionState*, *ActivityGraph*, etc.; e um elemento para gerenciamento dos modelos: *Package*.

O pacote *SPEM_Extensions* também está estruturado em sub-pacotes: *Basic Elements*, *Dependencies*, *Process Structue*, *Process Components* e *Process Lifecycle*. A figura 7 mostra essas relações de dependências.

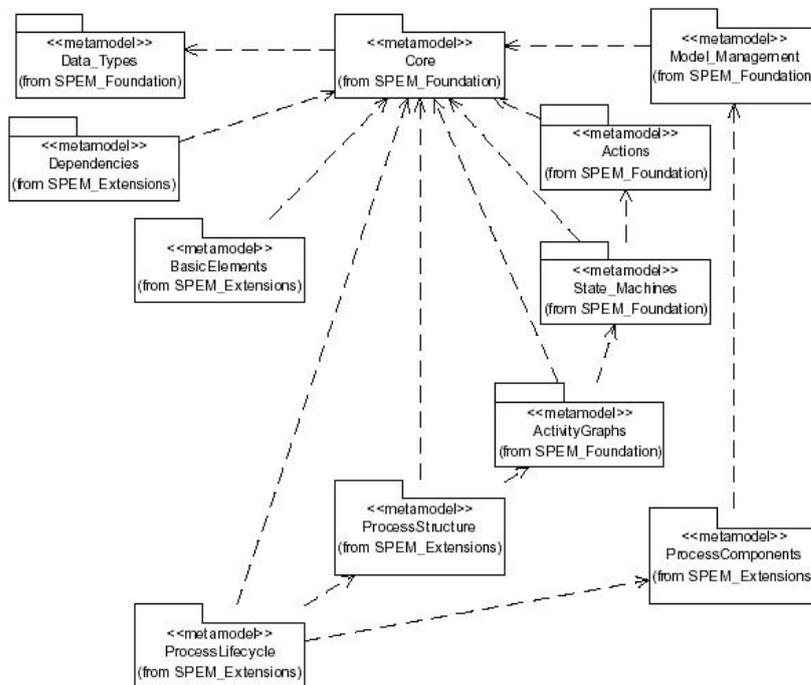


Figura 7- Detalhamento dos Sub-Pacotes do SPEM.

Comment [LA142]: Algumas figuras voce colocou numeração, ja outras nao, reveja isso.

O pacote *Basic Elements* (Elementos Básicos) contém elementos básicos para a descrição do processo. São elementos desse pacote:

- **Content Description:** contém a descrição de um elemento do processo;
- **Guidance:** responsável por fornecer informações mais detalhadas sobre os elementos do processo. Existem quatro tipos de Guidance: Technique, Checklist, Guideline e Template.

O pacote *Dependencies* contém elementos que representam dependências.

São elementos desse pacote:

- **Category:** serve para associar um pacote a um elemento de outro pacote;
- **Import:** serve para indicar que todo o conteúdo do pacote em questão é adicionado ao namespace do pacote origem;

Comment [LA143]: Excess de espaço

O pacote *Process Structure* contém os elementos principais para a construção da modelagem. São elementos desse pacote:

classe de produto e um Kind está relacionado ao tipo de produto.

- **Work Definition and Work Definition Parameter:** é um tipo de operação que representa o tipo de trabalho desenvolvido no processo, que pode ser uma Task, Step ou Activity.
- **Activity and Step:** é a principal subclasse de Work Definition e representa o trabalho realizado por um Role Definition. Uma atividade pode ser constituída por Steps que são elementos atômicos.
- **Role Use and Role Definition:** um Role Use representa o executor de uma ou mais definições de trabalho (Work Definitions) no processo. Role Definition representa papéis e responsabilidades em atividades (Activity).

O pacote *Process Components* contém elementos que têm por objetivo dividir o projeto em definições para que, assim, possam ser colocadas sob gerência de configuração e versão.

São elementos desse pacote:

- **Method Package and Process Package:** é um “recipiente” que pode possuir elementos de definição de métodos e de processo.
- **Process Component:** representa uma descrição do processo, sendo consistente e podendo ser reusado com outros componentes para criar um processo maior e mais complexo.
- **Process:** é definido a partir de um Process Component. Ele representa uma família de processos, dentro dos quais, mais processos podem ser definidos.
- **Category (instance):** é uma especialização de Category e divide as atividades do processo de acordo com um tema.

O pacote *Process Lifecycle* contém elementos que ajudam a definir a execução do processo, ou seja, como ele será executado. São elementos desse pacote:

- **Kind:** qualifica as entidades básicas. Por exemplo, um Guidance pode ser do Kind: White Paper, Guideline, Checklist ou Templates, e uma Activity como uma Activity (atividade) ou Phase (fase).
- **Precondition:** a cada Work Definition pode ser associada a uma pré-condição.
- **Postcondition:** semelhante a um Precondition, uma pré-condição pode ser associada a uma Work Definition.

4.2.1 OMG(Object Management Group)



Comment [a144]: Ver numeração da seção que seria 16.4.2

Comment [LA145]: O que quer dizer essa figura.

integração da empresa para uma ampla gama de tecnologias, incluindo: Real-time, e Especializadas em Embedded Systems, Analysis & Design, Arquitetura-Driven Modernização e Middleware e uma gama ainda maior de indústrias, incluindo: Modelagem de Negócios e da Integração, C4I, Finanças, Governo, Saúde, Conformidade Legal, Life Sciences Research, Manufacturing Technology, robótica, baseados em software de comunicações e espaço.

Padrões de modelagem da OMG permitem um design visual forte, com a execução e manutenção de software e de outros processos. Padrões de middleware OMG e perfis com base no objeto comum Request Broker Architecture (CORBA[®]) que dá suporte a uma ampla variedade de indústrias.[15]

Modelagem Padrões da OMG's, incluindo a Unified Modeling Language[™] (UML[®]) e Model Driven Architecture[®] (MDA[®]), que permitem um desenho com um visual forte, e a execução e manutenção de software e de outros processos, incluindo sistemas de TI Modelagem de Processos e Gestão Empresarial. Padrões de middleware OMG's e perfis com base no objeto comum Request Broker Architecture (CORBA[®]) que é suportado em uma ampla variedade de indústrias.

O documento de requisitos que inicia cada configuração padrão OMG-atividade (Request for Proposal) e outros documentos fundamentais estão disponíveis para visualização por qualquer pessoa, membro ou não. Email discussão, participação de reuniões, e votação é restrita aos membros, apesar de sócios potenciais são convidados a participar de uma reunião ou duas, como observador convidado.

OMG é também o produtor do evento para a Internacionalização. OMG vem produzindo eventos durante quinze anos e fornece uma gama completa de serviços de conferência de gestão e desenvolvimento de competências e conferência, produção e suporte a mensagens orientada é a promoção da mídia.. Além disso, produz atualmente quatro Reuniões Técnicas da OMG e nove conferências e workshops a cada ano para os seus membros, em locais espalhados pelo mundo.

Comment [a146]: Incluir também neste parágrafo SPEM e BPMN, vistos nas seções anteriores, como padrões da OMG.



Figura 8 (Companhias Ligadas a OMG)

6.1. Ferramentas de Modelagem

As diversas linguagens ou abordagens de modelagem de processos apresentam diferentes elementos de um processo [4], [6]. No entanto, alguns elementos são comuns às várias abordagens como: agentes ou atores, papéis, atividades e artefatos ou produtos. Os agentes ou atores são entidades que executam um processo e podem ser pessoas, sistemas ou ferramentas.

Um ator pode desempenhar vários papéis. Os papéis descrevem um conjunto de responsabilidades de atores ou grupos, os direitos e as habilidades necessárias para realizar uma atividade específica do processo de software. Uma atividade inclui e implementa os procedimentos, as regras, as políticas e os objetivos para gerar e modificar um conjunto de artefatos. Um artefato ou produto é o (sub)produto e a matéria prima de um processo. Os (sub)produtos podem ser criados, acessados ou modificados durante a atividade do processo.

Atualmente, existem diversas ferramentas de software disponíveis no mercado para suportar e capacitar esforços para a engenharia de processos de negócio. Exemplos típicos são:

- **Ferramentas BPR** (*Business Process Reengineering*)
Uma ferramenta BPR pode ser usada para modelar e analisar processos de negócio. A representação visual dos processos e a habilidade para avaliar alternativas suportam a engenharia de processos.
- **Sistemas ERP** (*Enterprise Resource Planning*)
Sistemas ERP automatizam processos de manufatura, organiza livros contábeis, e delimita departamentos corporativos. Uma representação explícita do processo de negócio é usada como ponto de partida para a configuração destes sistemas.
- **Sistemas WFM** (*Workflow Management*)
Um sistema WFM é uma ferramenta de software genérica, no qual permite a definição, execução, registro e controle de fluxos de *workflows*. Na essência, o sistema WFM é um bloco de construção genérico para suportar processos de negócio.
- **IBM Rational Process Workbench**: é uma ferramenta *construída como um Rational Rose add-in*.
- **IRIS Process automation suite** (www.osellus.com).
 - 100% compatível com **SPEM**

Comment [LA147]: Primeiro paragrafo nao ha recuo.

Comment [LA148]: Veja o excess de espaçamento.

Comment [LA149]: Recuo.

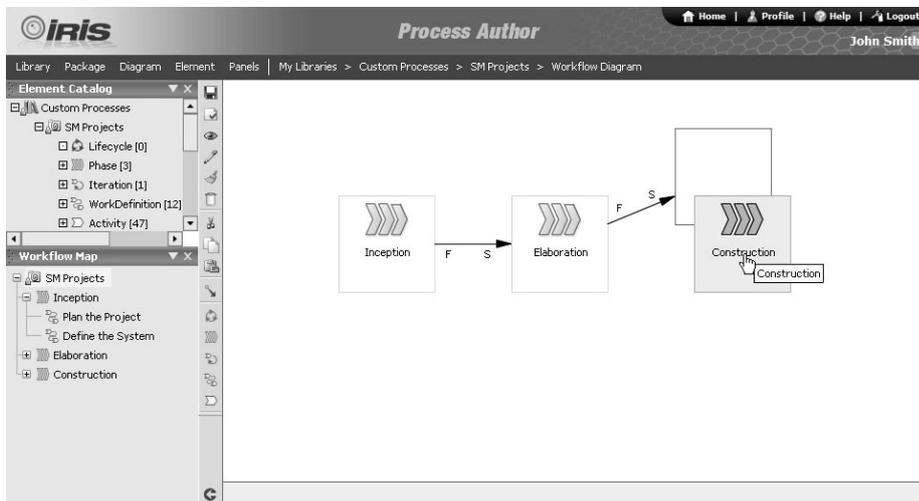


Figura x- Ferramenta IRIS

Comment [LA150]: Referencia da figura, e numeração.

- **Aris Platform**

- Lançada em 1992
- Principais clientes: OI, Petrobras, BNDES e VALE

- **Pontos forte**

- Excelente recursos de modelagem organizacional, capturando diversas dimensões do negócio(estratégia, organograma, produtos, processos, riscos, informações).
- Poderosos recursos de simulação, permitindo comparação de diferentes cenários.
- Repositório integrado de informação permite análises de impacto ricas (ex: pessoas que precisam ser treinadas).
- Componentes adicionais para BSC, Sarbanes-Oxley, Arquitetura Empresarial e Implantação SAP, entre outros.
- Poderosa ferramenta de inteligência, sugerindo melhorias nos processos.
- Grande número de profissionais treinados no mercado

- **Pontos de Atenção**

- Não é um BPMS, pois não possui motor de processos
- Métodos de exportação de informações para motores de processos são geralmente através de BPEL, o que gera forte ruptura semântica
- Suporte incompleto a BPMN

- *Eclipse Process framework*
- *JBoss jBPM*
 - *Lançado em 2006*
 - **Pontos Fortes**
 - Política baseada em *open source*
 - Oferece uma versão gratuita
 - O produto pode executar em uma infra-estrutura baseada em software livre
 - Está inserido em uma plataforma completa de middleware(jBoss Enterprise SOA Platform), que inclui servidor de aplicação, enterprise service bus e motor de regras de negócio, entre outros.
 - Pode ser uma opção para “process-enable” aplicações JEE.
 - **Pontos de Atenção**
 - É uma ferramenta voltada para desenvolvedores Java, não sendo adequada para profissionais de negócios, processos ou mesmo analistas de sistemas.
 - É baseada em notação proprietária
 - Não possui recursos de monitoramento da execução do processo

Enquanto que as ferramentas BPR suportam o “re-pensamento” de processos de negócio, os sistemas ERP e WFM são as aplicações de software que tornam a engenharia de processos possível. Cada uma dessas ferramentas requer uma representação explícita dos processos, sendo que a maioria das técnicas de modelagem existentes são utilizadas apenas por uma delas. Poucas ferramentas utilizam uma técnica genérica como redes Petri, SADT, IDEF e EPC

XX Ferramentas Para BPMN
BizAgi(BPMN)

XX Ferramentas Para SPEM
IRIS Process automation suite

6.1.1. Comparação entre as Ferramentas

IRIS	BizAgi
Notação utilizada	
SPEM	BPMN

Interface	
MUITO SIMPLES	COMPLEXA
Apoio ao usuário	
E-mail ou Tel	Fóruns
Reconhecimento / difusão da ferramenta	
Brasil, principalmente, e exterior	Mundialmente Difundida
Funcionalidades interessantes	
Publicação em HTML, exportação/importação entre ferramentas	Automação com BPM (Business Process Management)
Custo	
Acadêmico livre e completo com baixo Custo	Software livre com módulos pagos

7.1. Sugestões de Leitura

Business Process Management “Concepts, languages, Architectures”,
Mathias Weske.

Comment [a151]: Colocar texto descritivo , caso o leitor queira se aprofundar no assunto , para cada r sugestão de leitura

Building the agile enterprise with SOA, BPM AND MBM
FRED A. CUMMINS

Comment [LA152]: Uma breve descrição do que se trata, para chamar a atenção do leitor

Business Process Management Workshops
Arthur ter Hofstede
Boualem Benatallah
Hye-Young Paik(Eds.)

Enterprise Architecture for integration
Rapid, Delivery, Methods and Technologies
Clive Finkelstein

Exercício

- A modelagem tem o potencial para produzir ganhos que justificarão o tempo e o esforço previstos?
- A modelagem será estruturada de modo que as pessoas compreenderão suas saídas (não tão grandes e complexas que causem desânimo)?
- As pessoas entendem porque nós estamos modelando os processos, e o “o que isso pode trazer de benefícios para elas”?

*ex para BPMN / SPEM e Ferramentas

Referencias

[1] http://pt.wikipedia.org/wiki/Business_Process_Modeling

[2] <http://www.bpmn.org>

[SPEM, 2008] THE OBJECT MANAGEMENT GROUP, INC. Software Process Engineering Metamodel Specification (SPEM). Abril 2008.

[3] Object management Group -Software Process Engineering

Metamodel Specification Version 1.0, formal 02-11-14, Nov 2002.

[4] Acuña, S., Ferré, X., “Software Process Modeling”, Proceedings of The 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, Florida, USA.

1-6.

[6] Fuggetta, A., “Software Process: A Roadmap”, Future of

Software Engeneering, Limerick Ireland, 2000.

[7] SPEM 1.1 (<http://www.omg.org/docs/formal/05-01-06.pdf>)

[8] Wautelet Y., Kolp M. and Achbany Y.: S-Tropos, An Iterative SPEM-Centric Software Project Management Process, Working Paper IAG (2005).

[9] Eclipse Modeling Project (<http://www.eclipse.org/modeling>)

[10] Eclipse Process Framework Project (<http://www.eclipse.org/epf/>)

[11] SPEM 2.0 RFP ad/2004-11-04: 3rd Revised Submission (2006)

Comment [a153]: Retirar os colchetes das referencias e para cada referencia web, complementar com “acessado em “

- Também colocar as referencias em ordem alfabética.

Comment [a154]: Referencias [1] e [2], [15] colocar no padrão, p exemplo: BPMN , HTTP://www.bpmn.org, Acessado em outubro 2009.

■ HAMMER, M.; CHAMPY, J., *Reengenharia: revolucionando a empresa em função dos clientes, da concorrência e das grandes mudanças da gerencia*. Rio de Janeiro: Campus, 1995.

■ CHING, H. Y. *Gestão baseada em custeio por atividades*. 3ª ed. São Paulo: Atlas, 2001.

[12] Pinto Filho, J. B. A. (2007), "Gestão de Processos de Negócio: Uma adaptação da Metodologia de Rummler-Brache Baseada numa Aplicação Real", Dissertação de mestrado em ciências da computação, UFPE, Recife.

[13] GONÇALVES, J.E.L., *As empresas são grandes coleções de processos*. São Paulo: Revista de Administração de Empresas, Jan./Mar 2000, v.40.

[14] RUMMLER, G.; BRACHE, A., *Melhores Desempenhos das Empresas*. São Paulo, Makron Books, 1994

[15] <http://www.omg.org>.

Sumario...

Comment [LA155]: Todas suas referencias citadas no texto deve ser referenciada nas referencias bibliográficas.

Índice do Capítulo

6.1. INTRODUÇÃO	103
6.2. O QUE É QUALIDADE?	103
6.3. COMPETITIVIDADE X PRODUTIVIDADE	104
6.3.1. CONCEITO DE PRODUTIVIDADE	105
6.3.2. CONCEITO DE COMPETITIVIDADE	106
6.4. QUALIDADE TOTAL	108
6.4.1. DEMING	108
6.4.2. JURAN	109
6.4.3. CROSBY	109
6.4.4. FEIGENBAUN	110
6.4.5. ISHIKAWA	110
6.5. CONTROLE DA QUALIDADE TOTAL	112
6.5.1. APRESENTAÇÃO DO CONTROLE DA QUALIDADE TOTAL	113
6.5.2. SIGNIFICADO DO CONTROLE DA QUALIDADE TOTAL	114
6.5.3. PRINCÍPIOS DA QUALIDADE TOTAL	115
6.6. CONTROLE DE PROCESSO	119
6.6.1 CONCEITO DE PROCESSO	119
6.6.2 CONCEITO DE CONTROLE DE PROCESSO	119
6.6.3 MÉTODO DE CONTROLE DE PROCESSO	120
6.7. FERRAMENTAS DA QUALIDADE	121
6.7.1. AS SETE FERRAMENTAS DA QUALIDADE	121
6.8. GESTÃO DA QUALIDADE	126

6.8.1. GERENCIAMENTO POR DIRETRIZES	126
6.8.2. GERENCIAMENTO DA ROTINA	126
6.9. GARANTIA DA QUALIDADE	126
6.10. QUALIDADE NA INTERFACE COMPRAS/VENDAS	127
6.10.1. QUALIDADE NAS VENDAS	128
6.10.2. QUALIDADE NAS COMPRAS	129
6.11. IMPLANTAÇÃO DO TQC	131
6.11.1 FUNDAMENTOS	131
6.11.2 ORGANIZAÇÃO PARA IMPLANTAÇÃO	131
6.11.3 SISTEMA DE GERENCIAMENTO DA IMPLANTAÇÃO DO TQC	133
6.12 TÓPICOS DE PESQUISA	134
6.13 SUGESTÕES DE LEITURA	134
6.14 EXERCÍCIOS	134
6.15 REFERÊNCIAS	135

Capítulo

6

Controle da Qualidade Total

Flávia Leite Soares, Willame Pereira

<Incluir aqui, antes da primeira seção do capítulo, uma visão geral do capítulo.>

6.1. Introdução

Para entender a evolução da Qualidade nas organizações poderíamos iniciar traçando uma linha temporal, onde o marco inicial seria a criação de produtos segundo uma especificação técnica. Neste momento inicial, o foco dos gestores era oferecer produto ou serviço sem falhas, não eram os processos de gestão ou os clientes, internos ou externos. Qualidade, então, era oferecer um produto/serviço dentro do que foi especificado, era oferecer algo com ausência de defeitos.

A fim de atingir essa meta, o produto/serviço era verificado na medida exata da intensidade de inspeções realizadas. Ao longo do tempo, esse conceito tem mudado drasticamente, principalmente incorporando elementos relacionados ao cliente e a participação de toda organização, incluindo a comunidade onde ela na qual está inserida, os fornecedores, os acionistas e, principalmente, seu corpo funcional.

Obviamente a satisfação do cliente não é resultado apenas e tão somente do grau de conformidade com as especificações técnicas. O cliente não busca somente um produto sem defeito, mas também fatores como prazo e pontualidade de entrega, condições de pagamento, atendimento pré e pós-venda, flexibilidade, entre outros.

Nesse contexto, existem diversas abordagens que procuram tratar esses problemas por meio da implantação da Qualidade Total. O tema será apresentado neste capítulo com a introdução dos principais conceitos e métodos utilizados nas organizações para garantir que seus objetivos sejam alcançados.

6.2. O que é Qualidade?

O grande objetivo das organizações humanas é atender às necessidades do ser humano na sua luta pela sobrevivência na Terra. Um produto ou serviço de qualidade é aquele que atende perfeitamente, de forma confiável, de forma acessível, de forma segura e no tempo certo, às necessidades do cliente.

Segundo Armand Vallin Feigenbaum, qualidade "é a combinação de características de produtos e serviços de cada área da organização, para o atendimento das expectativas do cliente."

Para W. Edwards Deming qualidade "não é só ausência de defeitos. O consumidor é a parte mais importante da linha de produção. O verdadeiro critério da boa qualidade é a preferência do consumidor. É isto que garantirá a sobrevivência de sua empresa: a preferência do consumidor pelo seu produto em relação ao seu concorrente, hoje e no futuro".

Outra definição é a da FNQ - Fundação para o Prêmio Nacional da Qualidade - FNQ, baseada na NBR ISO 8052, que diz: "Qualidade é a totalidade das características de uma organização, atividade, processo ou produto de satisfazer as necessidades explícitas ou implícitas".

Segundo Paladini (2004), a qualidade pode ser desdobrada em dois planos bem definidos, sendo um "espacial" e outro "temporal", e que nesse contexto, a qualidade envolve muitos aspectos simultaneamente, ou

Comment [AV156]: Esta expressao ja foi usada antes.. sugestão: substituir por "nesse sentido" .. ou "nesse âmbito" ... ou outra parecida

seja, uma multiplicidade de itens, enquadrando-se no plano espacial; e sofre alterações conceituais ao longo do tempo num processo evolutivo, enquadrando-se no plano temporal.

Tendo em vista a complexidade e subjetividade do conceito de qualidade, e a variação de percepção da qualidade, Garvin (1992) desagrega o conceito de qualidade para analisar os elementos básicos. Em sua análise, o autor classifica ou categoriza as chamadas oito dimensões da qualidade, no intuito de dirimir as dúvidas e interpretações equivocadas sobre o assunto. As dimensões da qualidade segundo Garvin são: desempenho, características, confiabilidade, conformidade, durabilidade, atendimento, estética e qualidade percebida. Podemos observar que algumas desses elementos são técnicos, elementos que fazem com que o produto esteja dentro do especificado, exemplo: suas características e durabilidade. Outros elementos, como estética, confiabilidade e qualidade percebida, são abstratos, dependem da percepção do usuário. Estes elementos abstratos podem mudar de usuário a usuário do serviço, tendo cada um uma percepção diferente.

6.3. Competitividade x Produtividade

O assunto competitividade empresarial tem despertado um interesse cada vez maior, a partir principalmente da década de 70, quando empresas e nações que dominavam determinados setores do mercado foram destituídas de sua posição por concorrentes até então não levados em consideração.

Empresas aparentemente inexpugnáveis podem, devido às rápidas mudanças, ter sua sobrevivência ameaçada. Essas mudanças no posicionamento relativo das empresas não é algo pontual, mas sim uma tendência que se observa atualmente. Estratégias, técnicas e modelos de gestão revolucionários são instrumentos usados para tomar o lugar de empresas que durante décadas detinham grande fatia do mercado.

Um caso emblemático é o da tradicional indústria automobilística americana que, pela primeira vez na história, foi superada pelas montadoras asiáticas: elas vendem nos Estados Unidos mais do que as três maiores montadoras locais juntas.

É por este motivo que a preocupação atual da alta administração das empresas em todo mundo tem sido desenvolver sistemas administrativos (software) suficientemente fortes e ágeis de tal forma a garantir a sobrevivência das empresas.

Comment [AV157]: De onde vem essa estimativa? Seria interessante colocara a fonte.. tipow.. "segundo tal fonte.... vendem mais q .."

6.3.1. Conceito de Produtividade

Inicialmente, para sobreviver, uma empresa necessitava apenas produzir/fornecer produtos/serviços que fossem bons o suficiente para satisfazer ao cliente, ou seja, que tivessem qualidade.

Há muito tempo produzir/fornecer um bom produto/serviço deixou de ser um diferencial para ser uma obrigação. Para continuar no mercado, as empresas devem superar as expectativas dos clientes produzindo o melhor produto da forma mais eficiente possível, ou seja, serem produtivas.

A Organização Internacional do Trabalho (OIT) define produtividade como "a relação entre os bens e serviços produzidos e o valor dos recursos utilizados no processo de produção".

Já a definição de Burckminster Fuller passa a sensação de continuidade, da busca constante da otimização. "Produtividade é conseguir cada vez mais com cada vez menos".

Segundo Falconi, aumentar a produtividade é produzir cada vez mais e melhor com cada vez menos. Pode-se então representar a produtividade como o quociente entre o que a empresa produz (Output) e o que ela consome (Input):

$$\text{Produtividade} = \frac{\text{Output}}{\text{Input}}$$

As organizações, que têm como missão satisfazer às necessidades do ser humano, produzem produtos/serviços (Output) para atenderem às necessidades dos clientes.

Estes produtos/serviços devem ser especificados, projetados e produzidos de tal forma a terem valor, ou seja, serem necessários, desejados e ambicionados pelos clientes. O preço é função deste valor. Se a empresa for capaz de agregar muito valor por um baixo custo, ela dominará o mercado, pois os consumidores, evidentemente, sempre procurarão o máximo valor pelo seu dinheiro.

Este valor deve ser agregado ao menor custo (Input). O custo representa os valores que a organização retira da sociedade e aos quais agrega valor para esta mesma sociedade. Desta maneira, substituindo-se na equação anterior, output por Valor Produzido e Input por Valor Consumido podemos visualizar a produtividade como Taxa de Valor Agregado:

$$\text{Produtividade} = \frac{\text{Valor Produzido}}{\text{Valor Consumido}} = \text{Taxa de valor agregado}$$

Para aumentar a produtividade de uma organização humana, deve-se agregar o máximo de valor (máxima satisfação das necessidades dos clientes) ao menor custo. Quanto maior a produtividade de uma empresa, mais útil ela é para a sociedade, pois está atendendo às necessidades dos seus clientes a um baixo custo.

$$\text{Produtividade} = \frac{\text{Qualidade}}{\text{Custos}}$$

Esta definição de produtividade torna clara a afirmação do Prof. de Deming: “de que a produtividade é aumentada pela melhoria da qualidade e que este fato é de domínio de uma seleta minoria.”

Uma última definição de produtividade seria:

$$\text{Produtividade} = \frac{\text{Faturamento}}{\text{Custos}}$$

A vantagem da definição acima é que, além de levar em conta considerar todos os fatores internos da empresa (taxa de consumo de materiais, taxa de consumo de energia e taxa de utilização de informação), ela inclui também o cliente como fator decisivo de produtividade. Se o cliente não quiser estiver interessado em comprar, por maior que seja a eficiência da empresa, a produtividade cairá.

Para que tenham máxima produtividade, os três elementos básicos que constituem as organizações devem ser melhorados: equipamentos e materiais; procedimentos e métodos; e o ser humano.

Falconi propõe uma base conceitual para um programa de aumento de produtividade:

- É necessário fazer “aporte de conhecimento” de maneira a aumentar o ativo de conhecimento da empresa;
- Uma vez que existe uma limitação na velocidade do aprendizado humano, o aporte deve ser contínuo;
- É necessária uma postura voluntária para que o aprendizado ocorra de forma célere, portanto toda a empresa deve ser gerenciada de modo a elevar o moral das equipes;
- Dado que o ativo de conhecimento foi inserido nas pessoas, é importante criar condições para que a saída dessas pessoas da empresa seja evitada;
- Um programa de qualidade e produtividade é essencialmente um programa de “aporte de conhecimento” e para que ele seja realmente absorvido pela empresa é necessário tempo. Um programa de qualidade bem conduzido leva algo em torno de cinco anos. Os resultados são lentos e graduais, mas definitivos.

6.3.2. Conceito de Competitividade

Ser competitivo é ter a maior produtividade entre todos os seus concorrentes; é proporcionar algo único que seja valorizado pelo cliente. Quando a empresa consegue fornecer esse “algo único” a um preço aceitável pelo cliente e que ao mesmo tempo supere os custos adicionais, a empresa alcança um desempenho superior, pois, além de agregar o “valor da qualidade” ao seu produto ela pode se tornar única naquele mercado.

Como mostra a figura abaixo, o valor da qualidade é agregado aos produtos a partir dos três elementos básicos que constituem as organizações (equipamentos e materiais, procedimentos e métodos, e o ser humano).

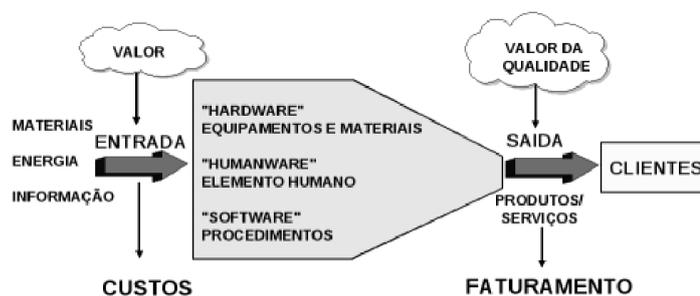


Figura 6.1. Agregação do valor da qualidade (Falconi)

Segundo W. Deming, “A produtividade é aumentada pela melhoria da qualidade. Esse fato é bem conhecido só por uma seleta minoria”. Apenas a minoria citada sabe que não existe um ponto de perfeição da qualidade, mas sim uma busca contínua pela qualidade ideal. As necessidades dos clientes e o mercado estão em constante mudança e, por isso, apenas as empresas que buscam a melhoria da qualidade estão aptas a sobreviver nesse novo cenário mundial de concorrência.

O que realmente garante a sobrevivência da empresa é a garantia de sua competitividade.



Figura 6.2. Fatores necessários à sobrevivência da empresa (Falconi)

O componente informação é de suma importância nesse cenário de competitividade, pois ela é [trata-se da](#) matéria-prima necessária à criação de conhecimento, um instrumento essencial para:

- Captar as necessidades dos clientes por meio de métodos e instrumentos sofisticados;
- Pesquisar e desenvolver novos produtos que melhor se adaptem àquelas necessidades;
- Pesquisar e desenvolver novos processos que garantam melhor qualidade de conformidade e custos mais baixos;
- Gerenciar sistemas administrativos que conduzam a maior produtividade;
- Comercializar e dar assistência técnica aos clientes.

6.4. Qualidade Total

Para se ter qualidade total dentro de uma organização, é necessário existir um controle total desta como um todo.

O Japão pós-guerra havia criado o TQC – Total Quality Control, que envolve todas as funções (fabricação, marketing, compras, etc). Porém houve a necessidade do envolvimento de todas as funções em níveis hierárquicos. O TQC evoluiu para o CWQC (Company Wide Quality Control – Controle Total por Toda Companhia).

O CWQC possui quatro tópicos principais:

- Envolvimento de outras funções no controle da qualidade além da fabricação;
- Participação dos empregados de todos os níveis hierárquicos;
- Filosofia de melhoria contínua;
- Forte interação com o cliente (quem define a qualidade);

O controle total da qualidade tem como premissa básica a satisfação da necessidade das pessoas, e conseqüentemente, o resultado desejado da empresa: Qualidade Total de todos os níveis e setores. Nas próximas seções serão [serão](#) descritas as abordagens para o Controle da Qualidade Total de acordo com alguns dos autores mais conhecidos.

6.4.1. Deming

Reconhecido mundialmente como o grande promotor do Controle da Qualidade no Japão, deixou grandes contribuições para o desenvolvimento da qualidade. [Deming \(1990\) desenvolveu](#) sua abordagem é baseada no uso de métodos estatísticos para reduzir custos e aumentar a produtividade e qualidade de produtos, Deming (1990). Para descrever sua filosofia, Deming definiu 14 pontos:

- 1) Criar uma constância de propósitos de melhorar produtos e serviços.
- 2) Adotar uma nova filosofia. É momento de iniciar um movimento por mudanças, de despertar para o desafio, de assumir a liderança em direção à transformação.
- 3) Eliminar a necessidade de inspeção em massa, priorizando a internalização da qualidade do produto. A qualidade não se origina da inspeção, mas do melhoramento do processo.
- 4) Acabar com o sistema de compras baseado apenas no preço. Minimizar o custo total por meio de relacionamentos duradouros, calcados na qualidade e na confiança com um único fornecedor para cada item.
- 5) Melhorar constantemente o sistema de planejamento, produção e serviços.
- 6) Implantar métodos de treinamento no local de trabalho.
- 7) Reformular métodos de supervisão; instituir a liderança com o objetivo de ajudar as pessoas a realizar um trabalho melhor.
- 8) Eliminar o medo da organização.
- 9) Romper as barreiras entre os departamentos.
- 10) Eliminar slogans, exortações e metas para a mão-de-obra que não ofereçam meios para alcançá-las.
- 11) Eliminar os padrões de trabalho e cotas numéricas.
- 12) Romper as barreiras que privam o empregado de ter orgulho do seu trabalho.
- 13) Estabelecer um programa rigoroso de educação e auto-aperfeiçoamento para todo o pessoal.
- 14) Criar uma estrutura na alta administração que tenha como função implantar os 13 pontos anteriores. A transformação é tarefa de todos.

O enfoque de Deming está no controle e melhoria de processo, não apresentando para tanto, um sistema estruturado, ou uma metodologia clara para a implementação de sua abordagem nas empresas.

6.4.2. Juran

Suas principais contribuições foram a definição e organização dos custos da qualidade e o enfoque da qualidade como estratégia empresarial. Juran atribui a responsabilidade pela qualidade final do produto ou serviço à função qualidade, que segundo Juran (1991) "é o conjunto das atividades através das quais atingimos a adequação ao uso, não importando em que parte da organização estas atividades são executadas."

A essência do *Juran Management System* para o gerenciamento da qualidade é denominada Trilogia de Juran e é constituída dos seguintes conceitos:

- Planejamento da Qualidade: processo de preparação para encontrar as metas de qualidade;
- Controle da Qualidade: processo de encontro das metas de qualidade estabelecidas durante as operações;
- Aperfeiçoamento da Qualidade: processo de melhoria contínua da qualidade por meio de mudanças planejadas, previstas e controladas.

A implantação desta abordagem é baseada na formação de equipes de projeto para a resolução de problemas, um a um, melhorando a qualidade continuamente. Não há, no entanto, uma preocupação em organizar todas as atividades da função qualidade, de modo a garantir os melhores resultados desde o início do processo.

6.4.3. Crosby

Sua abordagem baseia-se na prevenção. O pai da filosofia Zero Defeito, preconiza que a qualidade é assegurada se todos se esforçarem em fazer seu trabalho corretamente da primeira vez. Qualidade significa

conformidade com os requisitos e ela deve ser definida em termos quantitativos para ajudar a organização a agir com base em metas tangíveis.

Para Crosby (1985), a qualidade é responsabilidade dos trabalhadores, o autor não considera, no entanto, outros aspectos que afetem a qualidade e que estão fora do controle dos operários, como os problemas com a matéria-prima fornecida, erros de projeto e outros.

Para sedimentar sua filosofia Crosby (1985) instituiu seus 14 pontos, que constituem as etapas de implementação de sua abordagem, são eles:

- 1) Compromisso da alta gestão com a qualidade. Um documento com a definição da política de qualidade e os objetivos da empresa deve ser elaborado;
- 2) Constituição de equipes de melhoria da qualidade coordenadas pelos gerentes;
- 3) Medição dos resultados. Os indicadores devem ser introduzidos a fim de identificar as necessidades de melhoria;
- 4) Avaliação dos custos da qualidade;
- 5) Comunicação dos resultados aos supervisores e operários. Os funcionários devem compreender a importância do respeito às especificações e o custo das não conformidades;
- 6) Reunião para identificação dos problemas e tomada de ações corretivas;
- 7) Estabelecimento de um comitê informal para a divulgação do programa “zero defeitos” na organização;
- 8) Treinamento da gerência e supervisão para implementar o que lhes compete no programa;
- 9) Instituir “o dia zero defeitos”, onde os resultados anuais são divulgados e efetua-se o reconhecimento a todos os participantes do programa;
- 10) Estabelecimento dos objetivos a serem seguidos. Para transformar os compromissos em ação os indivíduos e os grupos devem ser encorajados a estabelecerem metas de aperfeiçoamento;
- 11) Eliminar as causas dos erros. Os empregados devem ser consultados sobre a origem dos problemas;
- 12) Recompensar aqueles que atingiram os seus objetivos;
- 13) Formar os conselhos da qualidade a fim de trocarem ideias e experiências;
- 14) Recomeçar e progredir sempre: faça tudo de novo.

A filosofia de Crosby é voltada mais para o comportamento humano como único meio para se garantir a qualidade. É necessário que haja meios bem definidos, através de uma metodologia bem estruturada, para garantir o sucesso do programa e a conquista da Qualidade Total.

6.4.4. Feigenbaum

Ficou conhecido pela introdução do termo *Total Quality Control* (TQC) em 1961: "um sistema eficiente para a integração do desenvolvimento da qualidade, da manutenção da qualidade e dos esforços de melhoramento da qualidade dos diversos grupos numa organização, para permitir produtos e serviços mais econômicos que levem em conta a satisfação total do consumidor".

Sua abordagem diz que a qualidade deixa de ser responsabilidade de um departamento especializado em controle da qualidade e passa a ser função de todas as áreas da empresa. Para Feigenbaum a Qualidade é uma filosofia de gestão e um compromisso com a excelência.

6.4.5. Ishikawa

A abordagem de Ishikawa nasceu a partir da compilação de diversos aspectos do trabalho de vários especialistas como Deming, Juran e Shewart, acrescentando a eles uma grande preocupação com a

Formatted: Font: Italic

participação do elemento humano e trazendo para o controle da qualidade uma visão humanística sob a influência dos trabalhos de Maslow, Herzberg e McGregor.

Sua filosofia é voltada para a obtenção da qualidade total (qualidade, custo, entrega, moral e segurança – conceitos que serão explicados na seção 6.5.2) com a participação de todas as pessoas da organização, da alta gerência aos operários do chão de fábrica. No TQC japonês, através de uma metodologia bem definida, todos os níveis empresariais colocam suas atividades diárias sob controle, garantindo a qualidade por toda a empresa.

Ishikawa (1993) enfatiza também a participação dos funcionários através dos Círculos de Controle de Qualidade (CCQ), para a melhoria contínua dos níveis de qualidade e resolução de problemas. Nos círculos de qualidade são destacadas as seguintes características:

- Voluntarismo: os círculos devem ser criados em bases voluntárias e não por ordens superiores.
- Autodesenvolvimento: os membros do círculo precisam ter vontade de estudar.
- Desenvolvimento mútuo: os membros do círculo precisam aspirar a expandir os seus horizontes e a cooperar com outros círculos.
- Eventual participação total: os círculos precisam estabelecer, como seu objetivo final, a participação total de todos os trabalhadores do mesmo local de trabalho.

A abordagem de Ishikawa, justamente por ser mais abrangente e conciliar diversas características das abordagens anteriores, acaba sendo criticada pela sua dificuldade de implantação, apesar de ser simples e clara. Na verdade, esta abordagem exige por parte da empresa um comprometimento e uma mobilização significativamente maiores do que nos outros casos. Em algumas vezes, chega-se a relacionar esta necessidade de persistência e entusiasmo com a cultura e tradição japonesas, o que também acaba gerando dúvidas quanto à validade desta abordagem em países ocidentais.

Tendo a necessidade de se optar por uma das abordagens apresentadas para desenvolver este trabalho, escolheu-se o TQC no estilo japonês, segundo a abordagem de Ishikawa. A escolha recaiu sobre esta linha por vários motivos, destacando-se os seguintes:

- Os pontos mais importantes das abordagens dos demais especialistas acabam aparecendo no TQC no estilo japonês, os quais podem ser verificados nos Princípios Básicos do TQC, que serão descritos neste capítulo, a partir da seção 6.5;
- Clareza da metodologia de implementação desta abordagem;
- Ampla divulgação desta abordagem entre as empresas brasileiras, principalmente através do Prof. Falconi Campos: Campos (1990, 1992, 1994).

Este trabalho tratará apenas do TQC no estilo japonês, procurando em alguns pontos demonstrar a coerência com as outras abordagens. É importante salientar que neste trabalho a metodologia do TQC possui grandes contribuições do Dr. Falconi Campos, que procurou adaptar alguns aspectos à cultura local, bem como, estruturar o sistema administrativo TQC em etapas bem claras para facilitar a sua implementação.

6.5. Controle da Qualidade Total

Campos (1992) define o TQC no modelo japonês como um sistema gerencial que, com o envolvimento de todas as pessoas em todos os setores da empresa, visa satisfazer suas necessidades, através da prática do controle da qualidade. Tendo como premissa básica que o objetivo principal de uma empresa é a sua sobrevivência, o TQC vai buscar isto através da satisfação das necessidades das pessoas.

Comment [abv158]: Não seria melhor deixar só a última data da referência?

Assim, o primeiro passo é identificar todas as pessoas afetadas pela sua existência, e como atender suas necessidades. Segundo o autor, de forma e em momentos diferentes, a empresa interage com consumidores, acionistas, empregados e com a comunidade na qual está situada. O quadro abaixo mostra como esta interação pode ocorrer:

Tabela 6.1. Satisfação das pessoas da empresa.

Pessoas	Meios
Consumidores	Qualidade de produtos e serviços, Alto Valor Agregado, Custo Proporcional.
Empregados	Remuneração justa, Condições de trabalho adequadas, Crescimento profissional e pessoal.
Acionistas	Produtividade → Lucratividade
Comunidade	Impostos → Geração de recursos, Meio Ambiente → Preservação, Geração de empregos.

O objetivo principal de uma empresa (sua sobrevivência por meio da satisfação das necessidades das pessoas) pode ser alcançado pela prática do Controle da Qualidade Total, que atende aos objetivos da empresa por ser um sistema gerencial com as seguintes características básicas:

- Parte do reconhecimento das necessidades das pessoas e estabelece padrões para o atendimento destas necessidades.
- Visa manter os padrões que atendem às necessidades das pessoas.
- Visa melhorar (continuamente) os padrões que atendem às necessidades das pessoas, a partir de uma visão estratégica com abordagem humanista.

Segundo Ernest & Young (1993), o TQC consiste na criação de uma vantagem competitiva sustentável, que se dá por meio do constante aprimoramento do processo de identificação e atendimento das necessidades e expectativas dos clientes quanto aos produtos e serviços requeridos, e da utilização eficiente dos recursos existentes de modo a agregar o máximo de valor ao resultado final. Os objetivos da utilização deste método gerencial são:

- Garantir uma maior satisfação do cliente, fornecendo produtos e serviços que correspondam às suas expectativas, monitorando suas constantes mudanças ("*customer in*");
- Melhorar a qualidade do atendimento;
- Maior eficiência e produtividade, mantendo cada etapa do processo produtivo sob controle, detectando possíveis falhas e rastreando suas causas;
- Maior integração do pessoal, promovendo a comunicação entre os vários setores e diferentes níveis hierárquicos (comunicação vertical e horizontal);
- Redução de custos, minimizando retrabalhos;
- Maior lucratividade e crescimento.

Para Ishikawa (1993), "praticar um bom controle de qualidade é desenvolver, projetar, produzir e comercializar um produto de qualidade que é mais econômico, mais útil e sempre satisfatório para o consumidor." De onde se pode concluir que a qualidade deixa de ser responsabilidade de um departamento de controle de qualidade para ser uma obrigação de todos: do presidente da organização ao funcionário do mais baixo nível hierárquico.

Formatted: Font: Times New Roman, Italic, Portuguese (Brazil)

6.5.1. Apresentação do Controle da Qualidade Total

O TQC, como é visto hoje, surgiu no Japão a partir de idéias americanas após a Segunda Guerra Mundial. O modelo apresenta contribuições de várias fontes, utiliza, por exemplo, alguns conceitos trazidos da escola da administração científica de Taylor, o controle estatístico do processo de Shewhart e as teorias humanísticas de Maslow, Herzberg e McGregor. Mas as maiores contribuições vieram de nomes como Deming, Juran e Ishikawa.

Deming deu um enfoque maior à utilização de métodos estatísticos de maneira sistemática. Juran por sua vez procurou mostrar que apenas o esforço da mão de obra no controle da qualidade não era suficiente, responsabilizando a administração por cerca de 85% dos problemas de qualidade. A busca da qualidade total passa a ser então uma função gerencial.

E Ishikawa é o responsável pela união de todos estes conhecimentos da maneira organizada e sistêmica como é conhecido o TQC hoje. Introduzindo, ainda, a participação de uma massa crítica de funcionários das empresas na resolução de problemas de qualidade com os chamados Círculos de Controle de Qualidade - CCQ.

Sabendo que os serviços possuem características bastante peculiares, pode-se entender a dificuldade em avaliar objetivamente a sua qualidade, assim como, quantificar as medidas de suas características. No entanto, diversas empresas do setor de serviços estão em estágio avançado de implementação do TQC, comprovando que o método é aplicável a qualquer ramo da indústria.

Cada empresa é diferente, mesmo atuando no mesmo ramo. Isto significa dizer que o método não é uma receita de bolo e que, em cada caso, deverão ser respeitadas as características próprias de cada empresa, como porte, número de funcionários e cultura da região. Enfim, o método permite que, caso a caso, as particularidades sejam analisadas e, quando necessário, sejam feitas adaptações.

6.5.2. Significado do Controle da Qualidade Total

Controlar uma organização humana significa detectar quais foram os fins, efeitos ou resultados não alcançados (que são os problemas da organização), analisar estes maus resultados buscando suas causas e atuar sobre estas causas de tal modo a melhorar os resultados.

Como o objetivo de uma organização humana é satisfazer as necessidades das pessoas, então o objetivo, o fim, o resultado desejado de uma empresa é a Qualidade Total que são todas as dimensões que afetam a satisfação das necessidades das pessoas e, por conseguinte a sobrevivência da empresa. Essas dimensões são mostradas na (Tabela 6.2.1) e tem o seguinte significado:

Tabela 6.2. Dimensões da Qualidade Total. Adaptado de [Campos 1992]

Dimensões da Qualidade Total	Pessoas Atingidas
QUALIDADE (de todos envolvidos)	Cliente, Vizinho
CUSTO (final e intermediário)	Cliente, Acionista, Empregado e Vizinho
ENTREGA (condições e indicadores)	Cliente
MORAL (satisfação)	Empregado
SEGURANÇA (empregados e usuários)	Cliente, Empregado e Vizinho

Comment [abv159]: Acho q pelo template as tabelas sao com fundo branco mesmo...

- **Qualidade:** diretamente ligada à satisfação do cliente interno ou externo. É medida por meio das características da qualidade dos produtos ou serviços finais ou intermediários da empresa. Ela inclui a qualidade do produto ou serviço (ausência de defeitos e a presença de características que irão agradar ao consumidor), a qualidade da rotina da empresa (previsibilidade e confiabilidade em todas as operações), a qualidade do treinamento, a qualidade da informação, a qualidade das pessoas, a qualidade da empresa, a qualidade da administração, a qualidade dos objetivos, a qualidade do sistema, a qualidade dos engenheiros, etc.
- **Custo:** não apenas o custo final, mas os custos intermediários. O preço é também importante, pois ele deve refletir a qualidade. Cobra-se pelo valor agregado.
- **Entrega:** condições de entrega dos produtos ou serviços finais e intermediários de uma empresa: índices de atrasos de entrega, índices de entrega em local errado e índices de entrega de quantidades erradas.
- **Moral:** mede o nível de satisfação de um grupo de pessoas. Pode ser medido de várias maneiras: índice de *turn-over*, absenteísmo, índice de reclamações trabalhistas, etc.
- **Segurança:** mede-se a segurança dos empregados e usuários do produto por meio de índices como número de acidentes, índice de gravidade, etc.

Portanto, se o objetivo é atingir a Qualidade Total, devemos medir os resultados para saber se este objetivo foi alcançado ou não. Diante de qualquer destes resultados (fins) que estejam fora do valor desejado, deve-se controlar (buscar causas e atuar).

$$TQC = (\text{Controle} + \text{Qualidade})\text{Total}$$

$$TQC = \text{Controle Total} + \text{Qualidade Total}$$

Controle total é o controle exercido por todas as pessoas da empresa, de forma harmônica (sistêmica) e metódica (baseado no ciclo PDCA). Qualidade total é o verdadeiro objetivo de qualquer organização humana: satisfação das necessidades de todas as pessoas. TQC é o controle exercido por todas as pessoas para a satisfação das necessidades de todas as pessoas.

Comment [W160]: Referenciar dizendo que isso vai ser dito lá na frente.

6.5.3. Princípios da Qualidade Total

O que está por trás do conceito de TQC, bem como de sua metodologia é uma filosofia muito bem definida. Segundo Campos (1992) as empresas que adotam o TQC como modelo gerencial seguem alguns princípios básicos:

1. **Orientação pelo cliente:** Produzir e fornecer serviços e produtos que sejam definitivamente requisitados pelo consumidor.

Faz parte do passado a época em que a demanda era muito maior que a oferta e, com isto, as empresas podiam fabricar seus produtos e serviços independentemente das necessidades dos consumidores. Tudo o que era produzido era consumido pela escassez de ofertas. Os consumidores, então, adaptavam suas necessidades em função do que podiam conseguir no mercado.

A conjuntura atual mudou: a demanda continua grande, mas a oferta multiplicou-se em número muito maior. Agora as empresas precisam adaptar-se aos gostos e necessidades dos clientes e quem não seguir essa tendência corre o risco de ficar fora do mercado.

As empresas atentas à nova realidade criam um canal de comunicação sempre aberto com o mercado promovendo uma contínua conversação. Este canal tem como função básica saber o que o cliente pensa em todas as etapas da compra do produto ou serviço.

O que o cliente precisa, quais são suas necessidades, o que ele espera do produto ou serviço e o que a empresa deveria estar oferecendo? O que ele espera da empresa durante a compra e qual deve ser a postura da empresa representada no momento da compra pelo funcionário de linha de frente? Qual sua impressão pós compra, se ele está satisfeito, sim, não, por quê?

Todas estas informações devem ser tratadas dentro da organização e para funcionar como ponto de partida para o desenvolvimento de novos produtos e serviços e implantação de novas tecnologias. Além disso, a empresa precisa ter uma infra-estrutura que garanta a ausência de erros em todas as etapas do processo produtivo até o cliente, instalando uma rede de serviços para total satisfação que deve ser melhorada continuamente.

2. **Qualidade em primeiro lugar:** Conseguir a sobrevivência por meio do lucro contínuo pelo Domínio da Qualidade.

Dando-se prioridade à qualidade, os lucros virão como consequência. Em Ishikawa (1993), o autor afirma que se uma empresa segue o princípio da qualidade em primeiro lugar, seus lucros aumentarão com o decorrer do tempo. Mas se uma empresa persegue o objetivo de atingir lucros em curto prazo, perderá a competitividade no mercado internacional e, a longo prazo, perderá os lucros.

Deming mostra como as coisas acontecem em uma reação em cadeia quando o foco da empresa está na qualidade.

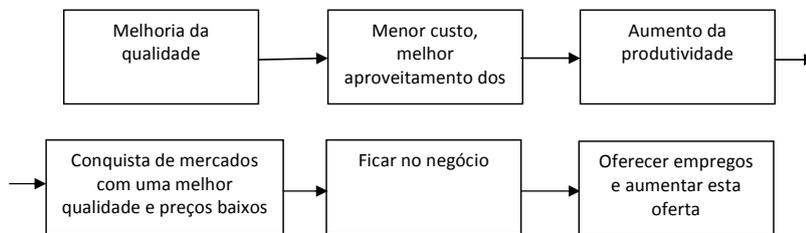


Figura 6.3. Cadeia Competitiva da Empresa. Walton (1989).

Field Code Changed

Neste contexto, a empresa deve adotar uma postura de preocupação constante com a qualidade de todos os processos da organização. Iniciando pela definição clara do que seria um produto ou serviço de qualidade com base nas necessidades e expectativas dos clientes e das possibilidades da empresa em questão. Em seguida, fazer um planejamento da qualidade, aliando neste planejamento o projeto/desenvolvimento de novos produtos/serviços e a garantia da qualidade da produção/prestação destes novos produtos/serviços.

Este princípio fomenta na empresa uma insatisfação contínua com os níveis de qualidade obtidos, buscando sempre alcançar níveis mais elevados.

3. **Ação orientada por prioridades:** Identificar o problema mais crítico e solucioná-lo pela mais alta prioridade.

A solução de problemas é iniciada pela identificação dos mesmos. A prática da maioria das empresas com uma gama de problemas aguardando soluções é a escolha aleatória ou com critérios restritos como, por exemplo, a simplicidade do problema em questão ou a grande soma de dinheiro envolvida. Estes critérios de seleção, no entanto, geralmente não levam em consideração os clientes envolvidos.

O TQC prega que os problemas da empresa sejam listados e, com base nas informações de clientes, mercado e diretrizes da alta administração, componha-se um ranking de prioridades. A análise e soluções destes problemas segue então a ordem de importância estabelecida, definindo-se metas a serem alcançadas e um cronograma a ser cumprido.

4. **Ação orientada por fatos e dados:** Falar, raciocinar e decidir com dados e com base em fatos.

Ainda hoje existem administradores que têm um conhecimento bastante abrangente e podem se dar ao luxo de resolverem todos os seus problemas apenas sabendo de sua existência, uma simples olhada e a solução já

está lá na ponta da língua. Acontece que nem todos são assim tão privilegiados, apesar de, em sua grande maioria, acharem-se capazes de resolverem tudo desta maneira tão simplista. O "achismo" continua a ser um método de auxílio a tomada de decisões muito utilizado. Os gerentes, supervisores, e funcionários em geral que possuem algum processo sob sua autoridade, devem habituar-se a trabalhar sempre com base em fatos e dados.

Muitas empresas, cientes desta necessidade, acostumaram-se a medir tudo, e anotar uma quantidade enorme de dados. Isto também não é desejável. A geração de dados por si só não resolve os problemas e deve ser feita de maneira planejada, ou seja, é imprescindível que seja feita uma correta identificação de quais são os dados realmente necessários, bem como, quais são os métodos e a frequência adequada de coleta. A partir destes dados, uma análise com base em técnicas estatísticas é que levará a resultados satisfatórios.

Comment [abv161]: Sugestao: substituit por "habituar-se"

Comment [abv162]: A partir daqui a frase fica contraditória. Melhor reorganiza-la para obter melhor coesão

5. **Controle de processos:** uma empresa não pode ser controlada por resultados, mas durante o processo. (O resultado final é tardio para se tomarem ações corretivas).

Para que o produto ou o serviço chegue ao cliente com qualidade assegurada, é necessário que todos na empresa estejam controlando seus processos, garantindo assim os resultados de seus trabalhos.

Este conceito se contrapõe à inspeção no final da linha, ou seja, na prestação do serviço ou na liberação do produto final, tão difundida no período pós guerra. No caso de serviços, este ponto é ainda mais importante.

Um produto defeituoso é encontrado antes de ser entregue ao cliente, gerando custos para a empresa, mas evitando o desencanto do consumidor. Já na prestação de serviços, o erro geralmente ocorre na presença do cliente impossibilitando a triagem de serviços bons e ruins.

Este princípio é fundamental para a implementação eficiente e eficaz do TQC em serviços. Deming dedica um de seus quatorze pontos para enfatizar a necessidade de eliminar a dependência da inspeção em massa: "Inspeção com o objetivo de encontrar peças defeituosas e jogá-las fora é tardia, ineficaz e cara. A qualidade não é fruto de inspeção, mas do aperfeiçoamento do processo", [Walton (1989)].

6. **Controle da dispersão:** Observar cuidadosamente a dispersão dos dados e isolar a causa fundamental da dispersão.

Os processos empresariais são afetados por vários fatores e cada fator é ainda influenciado por outros tantos, por isto a variabilidade dos processos é uma coisa até certo ponto esperada. No entanto, é necessário monitorar esta variabilidade dos processos, identificando pontos de controle que devem ser medidos. Os dados gerados são, então, analisados com ferramentas estatísticas com o objetivo de verificar como ocorre a distribuição dos dados e se a dispersão está ou não dentro de valores limites estabelecidos previamente. É ainda possível avaliar se as causas da dispersão são causas comuns (crônicas) ou causas especiais (ocorrem esporadicamente sem previsibilidade). Conforme os resultados, devem-se tomar as providências necessárias para manter os processos dentro de níveis aceitáveis de variabilidade.

7. **Próximo processo é seu cliente:** O cliente é um rei ou uma rainha com quem não se deve discutir, mas satisfazer os desejos, desde que razoáveis. Não deixe passar produto/serviço defeituoso.

Neste ponto surgem os conceitos de clientes e fornecedores internos. Estes conceitos são fundamentais tendo em vista a segmentação vigente nas empresas. É muito difícil encontrar um espírito de equipe que abranja os diversos departamentos; o mais comum é a rivalidade e a transferência de culpas e responsabilidades. Uma situação de companheirismo e ajuda mútua se desenvolve apenas onde encontra um clima organizacional receptivo, e isto é tarefa da alta administração. "É função da alta administração ajudar que se rompam as barreiras para que todos trabalhem em conjunto e em harmonia. É obrigação da alta gerência promover o trabalho em equipe", Mirshawka (1990).

Neste sentido, os objetivos maiores da empresa devem ser desdobrados para os diversos departamentos, cada departamento define então suas metas sempre levando em conta a empresa como um todo. As metas departamentais devem atender aos requisitos de seus clientes internos que são os processos posteriores, desta maneira forma-se uma cadeia de clientes e fornecedores dentro da organização. Assim, para que o cliente final

(externo) tenha suas necessidades atendidas é necessário que cada elo da cadeia seja fortificado por um relacionamento de parceria.

Segundo Ishikawa (1993), o controle de qualidade total não pode ser completo sem a total aceitação deste tipo de enfoque por todos os trabalhadores. O regionalismo precisa ser derrubado dando lugar à livre comunicação. O todo é sempre maior que a soma das partes se houver sinergia entre elas. Este é o espírito do TQC: trabalho em equipe com amizade, responsabilidade e respeito.

6.6. Controle de Processo

O controle de processo é a essência do gerenciamento em todos os níveis da empresa, desde o presidente até os operadores. O primeiro passo para entender o controle de processo é a compreensão do relacionamento causa-efeito [Campos 1992].

Sempre que algo ocorre (resultado, efeito, fim) existe um conjunto de causas (meios) que podem ter influenciado. Observando a importância da separação das causas de seus efeitos no gerenciamento e como, geralmente, esses dois termos são confundidos, os japoneses criaram o “diagrama de causa e efeito”. Este diagrama foi criado para que todas as pessoas da empresa pudessem exercitar a separação dos fins de seus meios. Mais sobre esse diagrama é explicado na seção de 6.7 Ferramentas da qualidade.

6.6.1 Conceito de Processo

Processo é um conjunto de causas (as quais provocam um ou mais efeitos). Um processo de desenvolvimento de software, por exemplo, possui várias causas, como requisitos, mão-de-obra, computadores, meio ambiente e métodos, que geram como resultado um produto de software.

O processo é controlado por meio dos seus efeitos. Caso o resultado do processo, citado no exemplo acima, tenha sido um software de má qualidade, então isso significa que algo de errado ocorreu durante o processo e que este precisa ser melhorado.

Sendo assim, para que se possa gerenciar cada processo é necessário avaliar (medir) os seus efeitos. O que se utiliza para avaliar os efeitos é chamado de itens de controle ou itens de controle de resultados. De acordo com Campos [Campos 1992], “os itens de controle de processo são índices numéricos estabelecidos sobre os efeitos de cada processo para medir a sua qualidade total”.

Portanto, um processo é gerenciado por meio de seus itens de controle que medem a qualidade, custo, entrega, moral e segurança (dimensões da qualidade total apresentadas na seção 6.5.2). Como exemplos de itens de controle de processos de software, podem-se citar a análise de segurança do produto e o nível de atendimento aos requisitos funcionais, dentre outros.

Numa empresa cada pessoa tem autoridade sobre o “seu processo” (meios), responsabilidade sobre os resultados (fins) deste processo e itens de controle. O item de controle é um dos pilares de um bom gerenciamento. Se não há itens de controle, não há gerência. Isso é a essência do gerenciamento participativo [Campos 1992].

6.6.2 Conceito de Controle de Processo

Para entendimento do conceito de “controle de processo”, supõe-se o seguinte caso: Em um novo aeroporto, a primeira coisa a ser feita é o **planejamento do processo**, que inclui as várias **metas** e vários **procedimentos-padrão** de pouso. Vários aviões aterrisam e decolam sem dificuldades **cumprindo os procedimentos-padrão**. Porém, certo dia um avião se acidenta durante os procedimentos de pouso. Ocorreu um desastre. Foi localizado um **problema** (resultado ou efeito indesejado do processo). Assim, a **causa** do problema passa a ser procurada. Como as causas podem ser várias (um conjunto de causas é um processo, como já dito anteriormente), procurar a causa é conduzir uma **análise de processo**.

Comment [W163]: Trocar depois esse exemplo por um ligado a software.

Depois de concluída a análise de processo e localizada a causa fundamental, é determinado um novo procedimento de pouso, de tal forma a garantir que a causa localizada seja evitada. O cumprimento desse novo procedimento de pouso por todos os aviões é chamado de **padronização**. Juntamente com essa padronização, novas **metas** e **itens de controle** são definidos visando evitar novos problemas.

Estas são as bases do controle de um processo. Manter um processo sob controle é saber localizar o problema, analisar o processo, padronizar e estabelecer itens de controle de tal forma que o problema não ocorra novamente. É importante observar que, após a análise de processo, o ato de padronizar e estabelecer itens de controle equivale a **replanejar o processo**. Replanejar sempre para melhores níveis.

Dessa forma, observa-se que o “controle de processo” consta de três ações fundamentais [Campos 1992]:

1) **Estabelecimento da “diretriz de controle” (Planejamento):**

Uma diretriz é estabelecida sobre os fins e meios de um processo e consta de:

- a) Meta, que é o nível de controle, ou seja, a faixa de valores desejada para o item de controle.
- b) Método, que são os procedimentos (meios) necessários para se atingir a Meta.

2) **Manutenção do nível de controle (Manutenção de padrões)**

Se todos os padrões estabelecidos na etapa anterior forem cumpridos, resultarão uma qualidade padrão, um custo padrão, uma entrega padrão, um moral padrão e segurança padrão. Sempre que ocorrem desvios deve-se:

- a) Atuar no resultado para repor imediatamente o processo em funcionamento. Por exemplo: queimou o motor – troca-se o motor.
- b) Atuar na causa para prevenir o reaparecimento do desvio. Por exemplo: queimou o motor – por que queimou o motor?

3) **Alteração da diretriz de controle (Melhorias)**

Como tudo muda constantemente, a diretriz de controle deve ser constantemente alterada de tal forma a garantir a sobrevivência do processo. Quando ocorre essa alteração, deve-se:

- a) Alterar a Meta, ou seja, alterar a faixa de valores proposta para o item de controle, alterar o nível de controle.
- b) Alterar o Método, ou seja, alterar os procedimentos-padrão de tal forma que o novo nível de controle seja atingido.

6.6.3 Método de Controle de Processo

Método é uma palavra de origem grega e é a soma das palavras META (que significa “além de”) e HODOS (que significa “caminho”). Portanto, método significa “caminho para se chegar a um ponto além do caminho”. Um ponto além do caminho pode ser um custo mais baixo, ou um ponto de superior qualidade, ou um ponto de melhor prazo de entrega, por exemplo.

Existe um “caminho” para isto que todos na empresa podem estudar e aprender, que é o método do Ciclo PDCA de controle. O PDCA é um método para a “prática do controle”. O verdadeiro sucesso comercial dos japoneses, nos últimos 15 anos, é fruto do uso do PDCA por todas as pessoas da empresa [Campos 1992].

6.6.3.1 O Ciclo PDCA de controle de processo

O controle de processo como citado na seção 6.6.2, é exercido por meio do Ciclo PDCA de controle de processos.

A Figura X mostra o Ciclo PDCA (PLAN, DO, CHECK, ACTION) composto das quatro fases básicas do controle: planejar, executar, verificar e atuar corretivamente.

Comment [W164]: Falta colocar a figura.

Os termos no Ciclo PDCA têm o seguinte significado [Campos 1992]:

- **Planejamento (P)** – Consiste em:
 - a) Estabelecer metas sobre os itens de controle;
 - b) Estabelecer a maneira (o método) para atingir as metas propostas;
- **Execução (D)** – Execução das tarefas exatamente como prevista no plano e coleta de dados para a verificação do processo. Nesta etapa é essencial o treinamento no trabalho decorrente da fase de planejamento.
- **Verificação (C)** – A partir dos dados coletados na execução, compara-se o resultado alcançado com a meta planejada.
- **Atuação corretiva (A)** – Esta é a etapa onde o usuário detectou desvios e atuará no sentido de fazer correções definitivas, de tal modo que o problema nunca volte a ocorrer.

6.7. Ferramentas da Qualidade

No TQC todas as decisões são tomadas com base em análise de fatos e dados. Para conseguir um melhor aproveitamento destes dados são utilizadas algumas técnicas e ferramentas adequadas. O objetivo principal é identificar os maiores problemas e através de análise adequada buscar a melhor solução.

O objetivo deste texto não é capacitar o leitor no uso destas ferramentas, mas inseri-las no contexto da qualidade total. Com este propósito serão apresentadas as ferramentas e uma breve descrição de sua utilização.

6.7.1. As Sete Ferramentas da Qualidade

São um conjunto de ferramentas estatísticas de uso consagrado para melhoria da qualidade de produtos, serviços e processos. A estatística desempenha um papel fundamental no gerenciamento da qualidade e da produtividade, por uma razão muito simples: não existem dois produtos exatamente iguais ou dois serviços prestados da mesma maneira, com as mesmas características. Tudo neste mundo varia e obedece a uma distribuição estatística. É necessário, então, ter um domínio sobre estas variações. A estatística oferece o suporte necessário para coletar, tabular, analisar e apresentar os dados destas variações.

As sete ferramentas da qualidade fazem parte de um grupo de métodos estatísticos elementares. É recomendado que estes métodos sejam de conhecimento de todas as pessoas, do presidente aos trabalhadores, e devem fazer parte do programa básico de treinamento da qualidade.

A. Folha de coleta de dados (verificação):

Consiste em uma planilha ou formulário para o registro de dados, no qual os itens a ser verificados já estão impressos ou definidos, de modo que os dados possam ser coletados de forma fácil e concisa. O objetivo desta ferramenta é gerar um quadro claro dos dados, que facilite a análise e tratamento posterior. Para tanto, é necessário que os dados obtidos correspondam à necessidade da empresa.

Três pontos são importantes na coleta de dados: ter um objetivo bem definido, obter contabilidade nas medições e registrar os dados de forma clara e organizada. As folhas de coleta de dados não seguem nenhum padrão preestabelecido, o importante é que cada empresa desenvolva o seu formulário de registro de dados, que permita que além dos dados seja registrado também o responsável pelas medições e registros, quando e como estas medições ocorreram. Outro fator imprescindível é que os responsáveis tenham o treinamento necessário para a correta utilização desta ferramenta.

Os erros mais freqüentes na elaboração do formulário são:

- Selecionar amostras tendenciosas;

- Falta de objetividade na definição do dado a ser observado;
- Colher dados insuficientemente ou em demasia;
- Erros na transcrição;
- Quando se apresentam dados graficamente, colocar muitas informações num mesmo gráfico (poluição visual);

Os dados podem ser coletados através questionários, folhas de verificação, checklist, etc. Dados como: frequência das observações, local e responsável são de extrema importância e não devem ser esquecidos.

FOLHA DE CONTROLE DE PROCESSO		FORMULÁRIO DE PESQUISA		
x x x x x x x	/ /	Perguntas	S	N
x x x x x x x				
x x x x x x x	/ /			
x x x x x x x	/			
x x x x x x x				

Figura 6. 4. Folhas de coleta de dados

B. Gráfico de Pareto:

Este método é utilizado para dividir um problema grande em vários problemas menores. O diagrama de Pareto elaborado com base numa folha de verificação ou de outra fonte de coleta de dados ajuda a dirigir a atenção e esforços para problemas verdadeiramente importantes. O diagrama parte do princípio de Pareto que busca separar os problemas vitais (poucos) dos triviais (muitos).

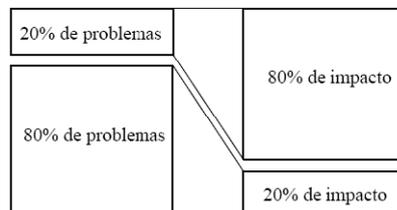


Figura 6. 5. Problemas x Impactos demonstrado pelo diagrama de Pareto.

Em síntese, o objetivo desta técnica é identificar as causas dos “poucos problemas vitais”, focando na solução dessas causas, dessa forma, eliminando uma parcela importante das perdas com um pequeno número de ações.

C. Diagrama de causa e efeito:

Este diagrama, também chamado de diagrama de Ishikawa ou espinha-de-peixe, é utilizado para mostrar a relação entre causas e efeito ou uma característica de qualidade e fatores. As causas principais podem ainda serem ramificadas em causas secundárias e/ou terciárias.

O diagrama de Ishikawa leva este nome em homenagem a seu criador, Kaoru Ishikawa, que desenvolveu esta ferramenta na década de 40. Ela se apresenta como uma ferramenta de qualidade muito

eficiente na identificação das causas e efeitos relacionados com a maioria dos problemas detectados em uma organização.

Uma grande seta indica o problema à direita. Ramos em formato de espinha de peixe representam as principais causas potenciais. Para um melhor resultado, todos os envolvidos devem participar da elaboração, para garantir que todas as causas sejam consideradas. Um coordenador deve ser nomeado e nenhuma idéia deve ser criticada, muito pelo contrário, deve-se estimular o intercâmbio de idéias para garantir uma visibilidade maior dos problemas e suas causas. As causas mais prováveis devem ser grifadas, mas todas devem ser analisadas.

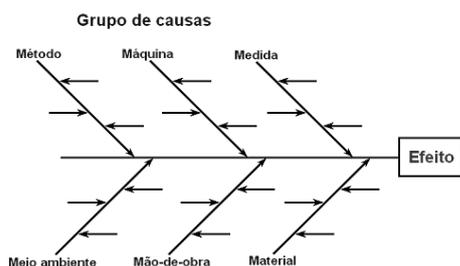


Figura 6.6. Aspecto de um diagrama de causa e efeito.

D. Fluxograma:

Esta técnica é utilizada para representar sequencialmente as etapas de um processo de produção, sendo uma fonte de oportunidades de melhorias para o processo, pois fornece um detalhamento das atividades concedendo um entendimento global do fluxo produtivo, de suas falhas e de seus gargalos. Os diagramas de fluxo são elaborados com uma série de símbolos com significados padronizados. É importante que os trabalhadores que confeccionem ou manipulem este tipo de diagramas conheçam a simbologia utilizada pela empresa.

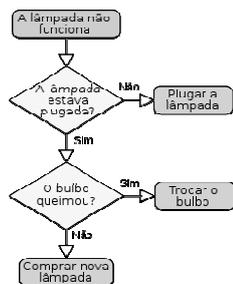


Figura 6.7. Exemplo de um fluxograma.

E. Histograma:

O histograma é um instrumento que possibilita ao analista uma visualização global de um grande número de dados, através da organização destes dados em um gráfico de barras separado por classes.

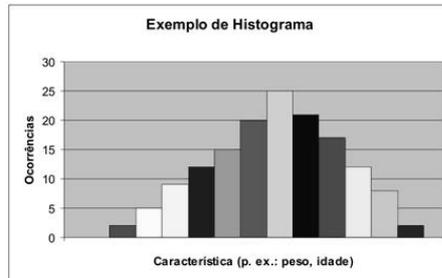


Figura 6.8. Aspecto de um histograma.

F. Diagrama de Dispersão:

O diagrama de dispersão é uma técnica gráfica utilizada para descobrir e mostrar relações entre dois conjuntos de dados associados que ocorrem aos pares. As relações entre os conjuntos de dados são inferidas pelo formato das nuvens de pontos formadas. Os diagramas podem apresentar diversas formas de acordo com a relação existente entre os dados.

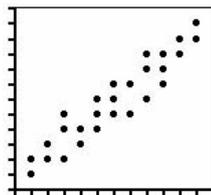


Figura 6.9. Aspecto de um diagrama de dispersão.

G. Gráfico de controle (acompanhamento):

O gráfico de controle é uma ferramenta utilizada para avaliar a estabilidade do processo, distinguindo as variações devidas às causas assinaláveis ou especiais das variações casuais inerentes ao processo. Ele fornece uma visão dinâmica do comportamento do processo, possibilitando uma verificação da evolução da sua qualidade. É simples e eficaz na fiscalização da variação do processo

As variações casuais repetem-se aleatoriamente dentro de limites previsíveis. As variações decorrentes de causas especiais necessitam de tratamento especial. É necessário, então, identificar, investigar e colocar sob controle alguns fatores que afetam o processo. Existe uma grande variedade de gráficos de controle entendendo a sua aplicação a todos os tipos de características mensuráveis de um processo.

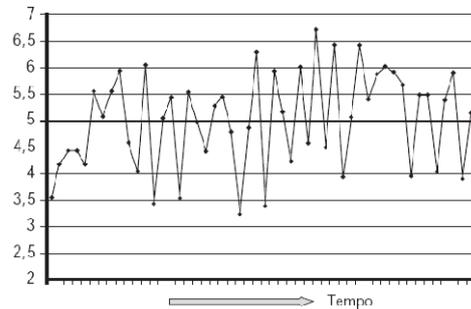


Figura 6. 10. Aspecto de um gráfico de controle.

6.8. Gestão da Qualidade

6.8.1. Gerenciamento por Diretrizes

6.8.2. Gerenciamento da Rotina

6.9. Garantia da Qualidade

A “Garantia da Qualidade” é uma função da empresa que visa confirmar que todas as atividades da qualidade estão sendo conduzidas da forma requerida, atendendo às necessidades do cliente (antecipando seus anseios) de forma completa e melhor que o concorrente. Por esse motivo de estar voltada a verificar continuamente se as necessidades do cliente estão sendo atendidas, a garantia da qualidade é considerada, segundo Campos [Campos 1992], como a “embaixatriz” do cliente na empresa.

A garantia da qualidade dentro do TQC é uma conquista; é um estágio avançado de uma empresa que praticou de maneira correta o controle da qualidade (via PDCA) em cada projeto e em cada processo. Dessa forma, uma empresa não poderá dizer que “instalou” uma garantia da qualidade pelo simples fato de ter estabelecido uma diretoria para este fim ou de ter satisfeito uma série de exigências [Campos 1992].

A garantia da qualidade busca o “defeito zero” (expressão citada na seção 6.4.3), ou seja, eliminar totalmente as falhas e só pode ser conseguida com a participação de todas as pessoas da empresa. A empresa pode ser vista como um processo constituído por vários processos menores, então cada pequeno processo da empresa deve garantir a qualidade para o processo seguinte, objetivando sempre a satisfação das necessidades do cliente interno. Sem esta participação voluntária e total das pessoas não se pode atingir a garantia da qualidade.

A qualidade é garantida por meio do planejamento da qualidade a ser colocada no mercado e pelo controle da qualidade conduzido por todas as pessoas da empresa (ciclo da garantia da qualidade). Além disso, deve ser periodicamente realizada uma auditoria da qualidade para verificar se todas as atividades da qualidade estão sendo conduzidas como planejado.

Sendo assim, a garantia da qualidade é constituída pelas seguintes etapas:

1. **Planejamento da qualidade.** Por meio de contato direto com o consumidor, no planejamento da qualidade são definidas as características da qualidade a serem agregadas ao produto ou ao serviço em cada processo interno, de forma a garantir a satisfação das necessidades do consumidor. Em cada processo, as características da qualidade do produto ou serviço que lhes são designadas são transformadas em itens de controle e gerenciadas.

Comment [abv165]: FALTA

Comment [W166]: Dizer em algum lugar que “empresa” pode significar uma organização sem fins lucrativos também ou qualquer outra entidade que implanta qualidade.

2. **Ciclo da garantia da qualidade.** Segundo Juran [Juran 1986], “a garantia da qualidade fornece proteção sob forma de avisos antecipados, que permitem a condução de ações corretivas antes do desastre. Quando se aplica este conceito para o desenvolvimento de produtos, a garantia da qualidade é chamada de garantia do projeto”. A garantia do projeto e do produto é feita através do ciclo de garantia da qualidade.
O ciclo da garantia da qualidade começa no cliente. Por meio de uma pesquisa de mercado, dados são coletados e classificados em necessidades de novos produtos e necessidades de melhorias em produtos existentes. Essas necessidades são enviadas para o desenvolvimento e são alinhadas com o planejamento da empresa. O projeto do produto é criado e depois a produção é iniciada, verificando sempre se a qualidade planejada está sendo seguida. Após a produção, o produto passa pela inspeção final e fecha-se o ciclo da garantia da qualidade, retornando ao cliente.
3. **Auditoria da qualidade:** A implantação do controle da qualidade em uma empresa precisa ser monitorada não só para verificar seus pontos fortes e fracos, mas também para orientar as pessoas e demonstrar o interesse contínuo da empresa pela qualidade [Campos 1992]. O produto da auditoria deve ser a orientação. Existem duas formas de auditoria: externa (conduzida pela organização do comprador, sendo realizada para certificação ou para obtenção de Prêmios Nacionais ou para outros fins) e interna (preparada pelo escritório de TQC).

De uma forma geral, pode-se dizer que garantir a qualidade é, conforme conceito japonês, garantir a satisfação do cliente por um longo tempo a um preço que este possa comprar e de forma melhor que os concorrentes. Satisfazer os clientes é atender a maior parte possível de suas necessidades (que mudam continuamente), no prazo certo, na quantidade certa e de forma segura para o cliente. Sendo assim, a qualidade só pode ser garantida se todas as pessoas da empresa praticarem o “controle da qualidade” de forma voluntária e motivada (Mais sobre motivação de pessoas pode ser lido no capítulo 13, Gestão de Pessoas). A garantia da qualidade deve ter como objetivo a sobrevivência da empresa na “guerra comercial” e não apenas satisfazer a algumas exigências de normas nacionais ou internacionais [Campos 1992].

6.10. Qualidade na Interface Compras/Vendas

Como foi já visto, para que a qualidade seja garantida numa organização, todos os processos devem garantir a qualidade para o processo seguinte. Então, os processos de relacionamento da empresa com seus clientes (vendas) e da empresa com seus fornecedores (compras) também devem ser norteados por esse mesmo princípio de garantia da qualidade.

6.10.1. Qualidade nas Vendas

O setor de vendas, dentro desse contexto de garantia da qualidade, não deve somente “receber pedidos” ou cumprir metas de vendas, algumas vezes até provocando a insatisfação do cliente. Esse setor deve então ser enriquecido e assumir novas responsabilidades, sendo melhor utilizar a denominação de marketing, que é mais envolvente e tem dentro de si a questão do “atendimento ao cliente”.

Campos [Campos 1992] cita uma frase de Ishikawa que define bem o marketing nesse contexto: “O marketing é a entrada e a saída da qualidade”. Ou seja, é por meio das atividades de marketing que será possível captar as necessidades e os anseios dos clientes e desenvolver novos produtos ou serviços que os satisfaçam.

Porém, em algumas empresas brasileiras, a conscientização do pessoal ligado ao setor de vendas sobre qualidade tem sido muito baixa. Nesses casos, o pensamento predominante é que o culpado pela falta de qualidade é a “produção” e as “reclamações devem ser feitas ao departamento de controle da qualidade, que é o responsável”. No entanto, no TQC, a qualidade é feita por todos e cada um é

responsável pela qualidade de seu processo. Sendo assim, o marketing é diretamente responsável pela qualidade do produto perante o consumidor [Campos 1992].

No TQC não se pode gerir a área de vendas apenas com base na “experiência” ou “sexto sentido”. Nesse caso, o controle tem que ser feito de forma racional, baseado em fatos e dados, análise de processo, divisão do processo total em segmentos gerenciáveis e giro completo do Ciclo PDCA em cada segmento. A Tabela 6.2 mostra exemplos de itens de controle de processo de marketing [Campos 1992].

Tabela 6.2. Exemplos de Itens de Controle nos Processos de Marketing. Adaptado de [Campos 1992]

Área	Itens de Controle de Processos de Marketing
Geral	<ol style="list-style-type: none"> 1. Educação e treinamento das pessoas do marketing em Controle de Qualidade. 2. Plano de vendas (obtenha precisão e análise das projeções). 3. Promoção de vendas (meça seus efeitos). 4. Quantidade vendida. 5. Lucro e despesas.
Clientes	<ol style="list-style-type: none"> 1. Atividade de <i>market-in</i> (satisfação das necessidades do cliente). 2. Educação e treinamento dos revendedores (coleta e compilação de dados referentes à qualidade). 3. Educação e treinamento do usuário do produto ou serviço. 4. Propaganda. 5. Cadastro dos clientes. 6. Reclamações. 7. Tempo de resposta da assistência técnica. 8. Nível de satisfação do cliente.
Informação e Análise	<ol style="list-style-type: none"> 1. Exatidão da informação em manuais, catálogos, propagandas, etc. 2. Nível de informação proveniente dos clientes. 3. Nível de informação relativo à confiabilidade do produto ou serviço. 4. Análise do preço de venda. 5. Mecanismo de informação à alta direção acerca dos problemas da clientela.
Produto ou Serviço	<ol style="list-style-type: none"> 1. Idéias de novos produtos ou serviços que irão atender às necessidades do cliente. 2. Participação no planejamento e desenvolvimento de novos produtos ou serviços. 3. Análise de segurança do produto ou serviço. 4. Custo no ciclo total de vida do produto ou serviço.
Estoque e Distribuição	<ol style="list-style-type: none"> 1. Estoque na fábrica (produtos e peças de reposição). 2. Taxa de atendimento imediato (produtos e peças de reposição). 3. Taxa de falta de estoque. 4. Taxa de produtos defeituosos no estoque. 5. Sistema de distribuição. 6. Custo de distribuição. 7. Perdas (de toda natureza, inclusive roubo), etc.

6.10.2. Qualidade nas Compras

Os métodos com que muitas empresas brasileiras atuam no setor de compras são inadequados sob a perspectiva da qualidade. Ainda se compra muito pelo menor preço, num relacionamento fornecedor/comprador que não prima, na maioria dos casos, pela confiança mútua.

De acordo com Campos [Campos 1992], uma primeira fase de conscientização no setor de compras seria reconhecer que o preço da matéria-prima adquirida é apenas um dos itens que compõem a qualidade. É evidente que se deve procurar pelo menor preço, mas contanto que isso venha acompanhado da boa qualidade do produto e confiabilidade dos prazos de entrega pelo fornecedor.

Uma segunda fase de conscientização seria objetivar o desenvolvimento de um relacionamento confiável e duradouro da empresa com seus fornecedores. Infelizmente, muitas empresas os maltratam, mudando suas programações de compra e prazos de pagamento ao sabor de suas necessidades, levando muitas pequenas e médias empresas brasileiras à falência.

O desenvolvimento dos fornecedores da empresa, para um novo tipo de relacionamento, é uma tarefa de longo prazo e que exige paciência antes de tudo. Campos [Campos 1992] cita os “Dez princípios” de Ishikawa que ajudam a montar um cenário futuro para o relacionamento fornecedor/comprador:

1. Ambos, fornecedor e comprador, são totalmente responsáveis pela aplicação do controle da qualidade, com entendimento e cooperação entre seus sistemas de controle da qualidade.
2. Ambos, fornecedor e comprador, devem ser mutuamente independentes e promover a independência do outro.
3. O comprador é responsável por entregar informações e exigências claras e adequadas, de tal maneira que o fornecedor saiba precisamente o que vai fabricar.
4. Ambos, fornecedor e comprador, antes de entrar nas negociações, devem fazer um contrato racional com relação à qualidade, quantidade, preço, termos de entrega e condições de pagamento.
5. O fornecedor é responsável pela garantia da qualidade que dará satisfação ao comprador, sendo também responsável pela apresentação dos dados necessários, quando requisitados pelo comprador.
6. Ambos, fornecedor e comprador, devem decidir com antecedência sobre o método de avaliação, de vários itens, que seja admitido como satisfatório para ambas as partes.
7. Ambos, fornecedor e comprador, devem estabelecer no contrato os sistemas e procedimentos por meio dos quais podem atingir acordo amigável de disputas, sempre que qualquer problema ocorrer.
8. Ambos, fornecedor e comprador, levando em consideração a posição do outro, devem trocar informações necessárias à melhor condução do controle da qualidade.
9. Ambos, fornecedor e comprador, devem sempre conduzir de maneira eficaz as atividades de controle dos negócios tais como pedido, planejamento de produção e estoque, trabalho administrativo e sistema, de tal maneira que o relacionamento deles seja mantido numa base amigável e satisfatória.
10. Ambos, fornecedor e comprador, quando estiverem tratando de seus negócios, devem sempre levar em conta o interesse do consumidor.

Para se obter qualidade nas compras, é importante destacar também que uma empresa não pode ser competitiva de forma isolada. Ela faz parte de uma cadeia de compradores/fornecedores que tem como objetivo final satisfazer as necessidades do consumidor. Ao comprar um produto de uma empresa, o consumidor, está, na verdade, comprando de uma “cadeia de empresas”. Então, é necessário que todas essas empresas busquem a qualidade de tal forma a tornar a “cadeia competitiva” [Campos 1992].

6.11. Implantação do TQC

6.11.1 Fundamentos

De acordo com Campos [Campos 1992], a implantação de um programa de qualidade é um processo de aprendizado e, portanto, não deve ter regras muito rígidas, mas estar adaptada às necessidades, usos e costumes da empresa. Um programa de qualidade deve ser visto como o aperfeiçoamento do gerenciamento existente.

Porém, existem alguns pontos básicos que devem ser seguidos para implantação do TQC [Campos 1992]:

- O TQC é implantando de cima para baixo (*top-down*). Esta implantação é assistida pelo “Escritório do TQC”.
- A implantação do TQC é de responsabilidade indelegável do Presidente da empresa. Se ele não perceber a necessidade do TQC, a sua implantação é impossível.
- A implantação do TQC é um processo de mudança comportamental e cultural e, portanto, é baseada num grande esforço de educação e treinamento. De acordo com Campos [Campos 1992]: “A educação é novo conhecimento para a mente e treinamento é a prática do uso do conhecimento. Só educar não resolve; é preciso educar e treinar. A prática é a mãe das mudanças”.
- Nunca implante o TQC sem orientação contínua de instituição qualificada e credenciada.

6.11.2 Organização para implantação

Conforme já dito anteriormente, o TQC é um programa *top-down*, portanto, o seu gerenciamento se inicia pelo presidente da empresa. Para isto o presidente indica um “Comitê de Implantação do TQC”, composto pelos seus principais executivos e presidido por ele. Participa também desde Comitê, o coordenador do TQC, que atua como secretário geral. A Figura 6.6 demonstra a organização para implantação do TQC.

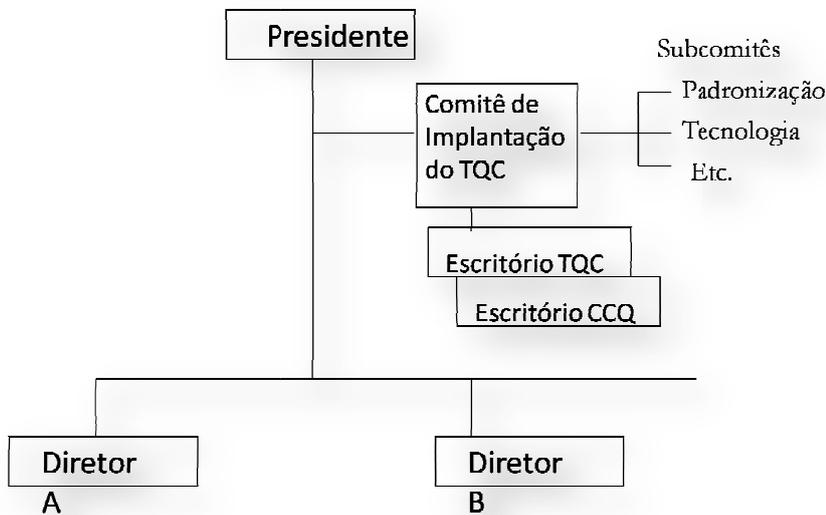


Figura 6.6. Organização para Implantação do TQC. Adaptado de [Campos 1992].

O Comitê de Implantação do TQC possui as seguintes funções:

- Avaliar e aprovar a proposição do Plano de Implantação do TQC feita pelo Coordenador do TQC.
- Acompanhar a evolução das “metas de sobrevivência” e o cumprimento do plano de implantação do TQC nas várias áreas da empresa.
- Recomendar ao presidente que atue nas causas dos desvios do plano, levantadas na reunião do Comitê.

O Escritório do TQC tem função de assessoria e consultoria interna. Ele deve centralizar todo o contato externo. É por ele que deve entrar todo o conhecimento sobre TQC. O chefe desse organismo é o Coordenador do TQC, o qual deve ser uma pessoa de confiança do Presidente e ter acesso fácil tanto ao Presidente como ao consultor externo.

Dentre algumas das muitas funções exercidas pelo Escritório do TQC, destacam-se:

- Propor a diretriz (metas + medidas) para implantação do TQC a ser aprovada pelo Comitê de Implantação.
- Promover a implementação do Conceito do TQC:
 - Propor o “Plano de Implantação do TQC”.
 - Executar a Educação e Treinamento para o TQC.
- Fornecer consultoria interna para o desenvolvimento gerencial dos chefes de seção e superiores.
- Coordenar a implantação das atividades de CCQ (Círculos de Controle da Qualidade, citados na seção 6.4.5).
- Monitorar todo o processo de implantação do TQC.
- Avaliar o estado atual e relatar mensalmente ao Comitê de Implantação do TQC.
- Difundir os resultados do TQC por toda a empresa.

6.11.3 Sistema de gerenciamento da implantação do TQC

O gerenciamento da implantação do TQC deve ser feito utilizando-se o Ciclo PDCA de controle de processos, como mostra a Figura 6.7.

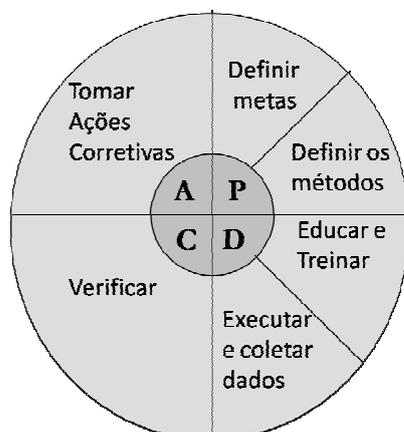


Figura 6.7 - Gerenciamento da Implantação do TQC. Adaptado de [Campos 1992].

Cada ponto deste processo gerencial é descrito nos itens a seguir:

- **Definir metas (Metas de sobrevivência):** As metas de sobrevivência são as que visam a garantir a sobrevivência da empresa à concorrência internacional. Essas metas decorrem da definição da

visão e estratégia da empresa. Um exemplo de meta poderia ser “Melhorar a qualidade do produto no prazo de um ano para sobreviver no mercado internacional”.

- **Definir os métodos:** Nesse ponto são definidos os métodos que permitirão atingir as metas propostas. Para isso, o “Plano de Implantação do TQC” deve ser montado pelo “Escritório do TQC” e apresentado como proposta ao “Comitê de Implantação do TQC” para análise e aprovação.
- **Educar e Treinar:** Nesse ponto, o Escritório do TQC deve propor o estabelecimento de um plano de educação e treinamento para o TQC e seu acompanhamento.
- **Executar e coletar dados:** O Escritório do TQC tem a responsabilidade de coletar dados sobre as “metas de sobrevivência” e sobre o andamento da implantação do plano em cada área. Estes dados são levados à reunião do Comitê de Implantação do TQC por meio de um relatório de progresso.
- **Verificar:** Nesse ponto é realizada a reunião do Comitê de Implantação do TQC onde são analisados as metas de sobrevivência e o relatório de progresso.
- **Tomar ações corretivas:** Aqui encerra o “ciclo de controle” da implantação do TQC. Nesse ponto, o Presidente vai autorizar a ação sobre as causas dos desvios do plano, recomendado pelo Comitê.

6.12 Tópicos de Pesquisa

- Análise da implantação do TQC em empresas de software e mensuração dos ganhos de produtividade obtidos.

6.13 Sugestões de Leitura

Campos, V. F. TQC - Controle Da Qualidade Total - No Estilo Japonês, QFCO, Belo Horizonte, 1992.
Ishikawa, K. Controle de Qualidade Total à maneira Japonesa, Rio de Janeiro, Campus, 1993.
Campos, V.F. Gerenciamento da Rotina do Trabalho do dia-a-dia', Rio de Janeiro, Bloch, 1994.

6.14 Exercícios

1) Dentre as opções, quais são fatores que influenciam na implementação de TQC?

I – Estratégia de qualidade bem elaborada

II – Grupo de comando para orientar a iniciativa

III – Treinamento apropriado

IV – Realizar inspeções necessariamente em todos os pontos da empresa

a) I e II estão corretas;

b) I, II e III estão corretas;

c) II, III e IV estão corretas;

d) Todas as alternativas estão corretas

2) A natureza seqüencial e contínua do processo de melhoramento contínuo é ilustrada pelo ciclo PDCA. PDCA significa:

a) Procurement, Design, Completion, Assurance

b) Plan, Do, Check, Act

- c) Produce, Design, Catalogue, Assess
 - d) Plan, Development, Check, Align
 - e) Produce, Deliver, Check, Assure
- 3) Sua filosofia é voltada para a obtenção da qualidade total com a participação de todas as pessoas da organização, da alta gerência aos operários do chão de fábrica.
- a) Feigenbaum
 - b) Crosby
 - c) Deming
 - d) Ishikawa
 - e) Juran
- 4) Qual dos princípios seguintes não é um princípio da qualidade total?
- a) Orientação pelo cliente
 - b) Ação orientada por prioridades
 - c) Controle de processos
 - d) Eficiência em primeiro lugar
 - e) Próximo processo é seu cliente
- 5) Um gráfico de sequência temporal mostrando valores plotados de uma estatística, incluindo linha central e limites de controle estatisticamente determinados é um(a):
- a) Carta de fluxo de processo
 - b) Diagrama de dispersão
 - c) Gráfico de corrida
 - d) Carta de controle
 - e) Diagrama de causa e efeito

6.15 Referências

- Campos, Vicente F. (1989) “Gerência da Qualidade Total”, Bloch Editores S.A. - QFCO, Rio de Janeiro.
- Campos, Vicente F. (1992) “TQC – Controle da Qualidade Total (no estilo japonês)”, QFCO, Belo Horizonte.
- Campos, Vicente F. (1992) “Qualidade Total - padronização de empresas”, QFCO, Minas Gerais.
- Campos, Vicente F. (1994) “Gerenciamento da Rotina do trabalho do dia a dia”, Bloch Editores S. A - QFCO, Rio de Janeiro.
- Crosby, Philip B. (1985) “Qualidade é investimento”, José Olympio, Rio de Janeiro.

- Deming, W. Edwards. (1990) “Qualidade a revolução da administração”, Marques Saraiva, Rio de Janeiro.
- Ernest & Young, Sotec. (1993) “Total Quality Management - a administração estratégica através da eficiência e qualidade em serviços”, Apostila.
- Garvin, D. A. (1992) “Gerenciando a qualidade: a visão estratégica e competitiva”, Qualitymark, Rio de Janeiro.
- Ishikawa, Kaoru.(1993) “Controle da qualidade total: à maneira japonesa”, Campus, Rio de Janeiro.
- Juran, J.M. (1986) “Management of Quality”. Juran Institute Inc, Wilton, CT, USA.
- Juran, J.M. (1991) “Controle da qualidade handbook - Vol. I”, Makron Books, São Paulo.
- Juran, J.M. (1991) “Controle da qualidade handbook - Vol. II”, Makron Books, São Paulo.
- Mirshawka, Victor. (1990) “A implantação da qualidade e da produtividade pelo método do Dr. Deming”, McGraw-Hill, São Paulo.
- Paladini, Edson P. (2004) “Gestão da Qualidade Teoria e Prática”, 2º. Ed, Atlas, São Paulo.
- Walton, Mary. (1989) “O método Deming de administração”, Marques Saraiva, Rio de Janeiro.

Capítulo

7

Normas ISO para Qualidade de Processos de Software

Hugo Vieira Lucena de Souza

O conceito de Gestão de Qualidade vem aos poucos sendo perfilhado como selo de reconhecimento para novos métodos, modelos e técnicas que melhoram expressivamente as perspectivas traçadas em projetos para várias organizações. A descrição bem elaborada dos processos, tão quanto suas aplicações e manutenções, englobam vários fatores que exigem um bom conhecimento das necessidades apresentadas, a identificação das principais atividades que os formam, como também as principais tarefas que motivarão um bom fluxo de funcionamento na busca para a implantação da qualidade.

Adjacente a este conceito enquadra-se um conjunto de normas internacionais provenientes da *International Organization for Standardization (ISO)*, distribuídas especificamente em vários campos da Engenharia de Software, responsáveis por avaliar e certificar características de processos e produtos, firmando assim garantia e segurança no desenvolvimento de sistemas de informação. Dentre estas normas destacam-se a série ISO 9000, com os requisitos mínimos para implantação e avaliação de um Sistema de Gestão para Qualidade (SGQ), a ISO/IEC 12207, responsável por ditar os processos mínimos essenciais para projetos em organizações, e a ISO/IEC 15504, responsável por nortear todos os processos utilizando-se de modelos de referência e medição para facilitar o desenvolvimento dos mesmos e suas etapas componentes.

Neste capítulo serão apresentados os conceitos relativos a normas técnicas e suas funções, os órgãos normativos que administram e publicam estes documentos, a série ISO 9000 com suas versões e perspectivas de qualidade adotadas em cada uma delas, as certificações ISO 9001 com seus princípios, estruturas e requisitos para Sistemas de Gestão de Qualidade, com foco principal para a ISO 9001:2008 e o guia de referência ISO/IEC 90003 destinado a projetos de sistemas em fábricas de software, além de apresentar as normas ISO/IEC 12207 e ISO/IEC 15504, com suas estruturas, diretrizes, restrições e descrições que são relacionadas para possibilitar uma melhor administração e implantação de melhorias nos processos de software.

Comment [j167]: Plural?

7.1. Conhecendo as normas

A utilização de normas foi algo que teve constantes evoluções ao longo dos anos. Desde a idade média, os filósofos padronizavam medidas e células nos primeiros documentos

Comment [j168]: reescrever

apresentando-o a sociedade enfatizou a importância em qualificar quaisquer produtos ou serviços com definições de suas principais diretrizes e restrições [ISO 2007].

O estabelecimento de modelos padrões para serem seguidos contribui com fatores de grande importância em todo o mundo. **Seja em proporções pequenas ou grandes**, a diferenciação qualitativa que pode ser obtida com a implantação de regras específicas serve como base para elaborar, ou mesmo melhorar legislações específicas para organizações, independente de tamanho e área relativa de abrangência.

Esses modelos de documentos, intitulados normas, são descritos como textos técnicos que buscam fixar padrões regulamentadores garantindo a qualidade de um produto de procedência industrial, a racionalização da produção, transporte e consumo de bens, a segurança das pessoas, a uniformidade dos meios de expressão e comunicação sendo aprovados por organismos reconhecidos que buscam validar comuns conhecimentos para utilização determinada de processos cuja correspondência seja satisfatória [FERREIRA 2004].

A ISO (2007) afirma **ainda** que *“pode-se também incluir ou tratar exclusivamente com a terminologia, símbolos, embalagem, marcação ou rotulagem, uma vez que se aplicam a um produto, processo ou método de produção.”*

A criação, edição, monitoramento e publicação, além de várias atividades que verificam e validam as normas são realizados através de vários processos hierárquicos classificados como *Work Draft* (esboços gráficos), por instituições colaborativas denominadas órgãos normativos [KOSCIANSKI e SOARES, 2007].

Comment [j169]: inverter os adjetivos, e colocar substantivo no final

Comment [j170]: ocultar

Comment [j171]: tirar ,

7.2. Organismos normativos

Para haver um controle unificado e evitar formação de grupos e comitês distintos, a hierarquia dos órgãos foi distribuída tomando por base os aspectos geográficos, facilitando a modificação e atualização [ISO 2009a]. De abrangência internacional, nacional ou mesmo regional, a criação de instituições normativas contribuiu muito na evolução e expansão para o uso de normas, deixando a sociedade consciente de que qualidade não é um componente complementar, mas sim indispensável.

A visão que se tem hoje sobre estas instituições desprende-se de um velho conceito de que a normatização referia-se apenas **a** manuais de instrução e leis de regulamentação impostas por governos, desmistificando as empresas qualificadas e alinhando uma concorrência mais justa levando em conta as condições de seus produtos vendidos [MUSSI e FERREIRA 1988].

Mussi e Ferreira (1988) afirmavam que *“A atuação desses organismos passou a ter uma relevância mais significativa quando as normas começaram a tratar aspectos de interesses comerciais e principalmente do relacionamento internacional”*.

Atualmente existe uma grande quantidade de organismos normativos espalhados pelo mundo. Grande parte deles aborda assuntos que condizem normas técnicas e normas de procedimentos relacionadas à avaliação de qualidade, como por exemplo, a ISO 9001, ou para características naturais destinadas ao meio ambiente, como por exemplo, a ISO 14000¹.

Comment [j172]: reescrever

Comment [j173]: tirar crase

A principal instituição de expressão considerável que define os padrões técnicos aplicáveis para normas está localizada na Europa. A ISO é regulamentadora de grande parte das normas existentes, caracterizando-se por impor o padrão de autenticidade de documentos para aplicação e validação das normas pelas demais instituições normativas.

7.2.1 ISO

À medida que vários padrões conceituais sobre determinados assuntos foram surgindo, a documentação para reconhecimento de suas funcionalidades também foi sendo elaborada. Em 1947, a fundação da *International Organization for Standardization* (ISO) em Genebra na Suíça foi um marco para o desenvolvimento mundial em relação às perspectivas de transformação que o mundo viria a passar a partir da década de 50 [ISO 2009a].

Com o intuito de conceder um controle para os documentos de normas, essa entidade ganhou relevante importância e respeito ao longo de sua história. Entre a data de sua fundação até os dias atuais, a publicação de aproximadamente 17500 padrões internacionais [ISO 2009b] para áreas como ciências exatas, saúde e humanas, transforma o pensamento de organizações, empresas e órgãos governamentais em 162 países² dos cinco continentes, que aos poucos utilizam as normas com uma visão mais coerente e realista sobre as necessidades de investimentos que precisam ser integradas nas organizações para que o diferencial qualitativo alcançado se torne um fator prioritário de negócios.

Mesmo detentora do controle das normas, a ISO adentrou-se de parcerias com outras instituições. Grande parte das normas publicadas pelo órgão parte de projetos conjuntos com instituições regulamentadoras de áreas específicas, assumindo assim a ISO, o papel de apenas registrar e apresentar como padrão o documento elaborado pelo comitê responsável por determinado campo de conhecimento.

7.2.2 IEC

No campo da tecnologia, grande parte das normas publicadas está subsidiada a parcerias realizadas com o *International Electrotechnical Commission* (IEC). Fundado em 1906 em Londres, Reino Unido, o órgão tornou-se o principal responsável para padronizar documentos, editoriais e normas que englobam características para sistemas elétricos e eletrônicos, nanotecnologias, multimídia, telecomunicações, além de simbologias determinadas especificamente para áreas como Engenharia Elétrica, Engenharia Eletrônica e Engenharia da Computação [IEC 2009a].

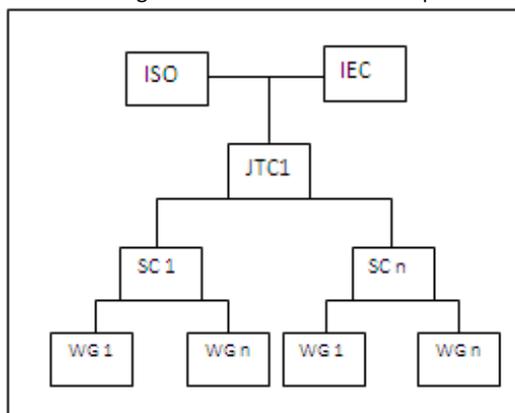
Os principais objetivos do órgão conforme o IEC (2009b) relacionados à Tecnologia da Informação são:

“Avaliar e melhorar a qualidade dos produtos e serviços abrangidos pelas suas normas, estabelecendo condições para a interoperabilidade dos sistemas complexos aumentando a eficiência dos processos industriais”.

Segundo o comitê, grande parte dessas melhorias se deve ao fato da dependência existente na concepção de produtos, com exigências que descrevem garantias de confiabilidade, desempenho e segurança.

A implantação de qualidade na Tecnologia da Informação foi algo que surgiu com a junção das normas ISO/TC 97 (*Information Technology*) e IEC/TC 83 (*Information Technology*) em 1987 [IEC 2009c]. A partir do projeto intitulado *Joint Technical Committee 1* (JTC1), a ISO e o IEC criaram um comitê responsável para proporcionar um melhor controle de criação, adequação e atualização de normas relacionadas à qualidade para Tecnologia da Informação.

A Figura 7.1 ilustra a atual hierarquia formada pela ISO, IEC e JTC1.



Comment [j174]: ocultar

Comment [j175]: :?

Comment [j176]: É isso mesmo?

Figura 7.1: Estrutura ISO/IEC/JTC1

Fonte: Adaptado de (Koscianski e Soares 2007)

Observando a Figura 7.1 nota-se que o JTC1 subdivide-se em partes menores chamadas *Sub Comissions* (SC). Cada subcomissão formadora do JTC1 é responsável por administrar um contingente de normas relacionadas a uma determinada área da Tecnologia da Informação, como por exemplo, Redes de Computadores, Banco de Dados, Arquiteturas e Sistemas Operacionais, dentre outras áreas diversificadas que complementam o ciclo de estudos sobre T.I. . Cada Subcomissão subdivide-se mais ainda em *Work Groups* (WG), que são grupos de estudos formados por profissionais de diversas corporações, sendo alguns deles eleitos ou nomeados, associações normativas internacionais e membros certificadores de tecnologias.

Para a Engenharia de Software, a subcomissão responsável é a de número sete (SC 7). Nesta comissão estão inclusos grupos relativos à padronização para documentação de software (WG 2), ferramentas de ambiente e desenvolvimento (WG 4), gerência, administração e gestão de processos (WG 10) gerência para qualidade de sistemas (WG 23), gerência de serviços para sistemas (WG 25), dentre outros grupos que compõem a comissão de certificação para assuntos relacionados à Qualidade de Software [JTC1 2008].

7.2.3 Organizações associadas internacionais

A regulamentação imposta pela ISO serve como base para um constante fortalecimento de propostas para o surgimento de novas normas internacionais. Os comitês e associações internacionais exercem um papel semelhante no âmbito de desenvolver pesquisas e projetos com o intuito de apoiar a normatização em uma dita assessoria de rigidez exercendo o controle necessário sobre as possíveis normas que estejam por vim.

Na Europa, por exemplo, órgãos como o Comitê Europeu de Normalização (CEN), o Comitê Europeu de Normalização Electrotécnica (CENELEC) e o Instituto Europeu para Normas de Telecomunicações (ETSI/IENT) regulamentam o padrão europeu de qualidade e segurança, além do funcionamento das associações européias inspecionando respectivos quadros de normas.

Para a América do Norte, o *American National Standards Institute* (ANSI) é um das principais associações internacionais atuais responsáveis pelas padronizações, localizado nos Estados Unidos. Na América Latina, a normatização fica a cargo da Associação Mercosul de Normalização (AMN) e da Comissão Paranaamericana de Normas Técnicas (COPANT), que deliberam os padrões de comercialização e adequação de serviços e produtos entre os países que formam o Mercado Comum do Sul (MERCOSUL).

No Brasil o controle normativo fica a cargo da Associação Brasileira de Normas Técnicas (ABNT). A instituição provém importantes projetos tecnológicos no Centro de Informações Tecnológicas (CIT) com o intuito de fornecer total apoio as empresas, profissionais da área, professores, estudantes entre outros que tenham interesse na área de normas técnicas que se desenvolve no Brasil e no exterior [ABNT, 2009a]. Assim como as demais organizações que regem padrões, a associação também é formada por diversos comitês (Comitês Brasileiros) e grupos de estudos específicos. O Comitê responsável pela verificação e adequação da qualidade é o Comitê Brasileiro vinte e cinco (CB-25), com perspectivas voltadas preferivelmente para Gestão da Qualidade, Garantia de Qualidade e para Avaliação da Conformidade para produtos e serviços [ABNT 2009b].

Comment [j177]: ocultar

Comment [j178]: espaço

Comment [j179]: mudar esse tópico, reescrever título

Comment [j180]: vir

Comment [j181]: acho desnecessário, focar em software

Dentre as demasiadas normas criadas e publicadas ao longo dos anos, nenhuma série de documentos obteve tanto destaque quanto a série ISO 9000. Spinola (2005) destaca a importância e o impacto desta série da seguinte forma:

“A série ISO 9000 de normas para gerenciamento de qualidade foi a que mais se desenvolveu em todos os tempos”.

Utilizadas em várias organizações diferenciadas, que buscam aperfeiçoar suas técnicas de produção e manutenção, a implantação de seus requisitos nos processos da organização possibilita um avanço comercial e empresarial consideravelmente positivo no que se diz respeito à melhoria interna e externa das atividades que os formam.

Camfield e Godoy (2003) afirmam ainda que as normas da série ISO 9000 integram valores indispensáveis para a implantação de melhoria contínua, sendo esta a principal dificuldade encontrada por várias organizações de diferentes portes. O estímulo de diferenciação que a série impõe nos seus documentos desenvolve temáticas de gestão bastante relevantes, no intuito de possibilitar a implantação e manutenção das atividades e tarefas de maneira mais sistemática e segura possível.

Não somente com a nomenclatura “9000” em seus títulos, esta família de normas desencadeia inúmeros documentos de vocabulários, documentos de requisitos como também guias técnicos diferenciados. A tabela 7.1 apresenta a família ISO atual:

Tabela 7.1: A família ISO 9000

Fonte: [Adaptado de MELLO et al. 2009]

Normas e diretrizes	Propósito
ISO 9000 – Sistemas de gestão da qualidade – Fundamentos e vocabulários	Estabelece termos e definições utilizados na família ISO 9000 para evitar interpretações errôneas durante seu uso.
ISO 9001 – Sistemas de Gestão da qualidade	Norma com fins contratuais utilizada para avaliar os sistemas em funcionamento para a busca de qualidade nas organizações.
ISO 9004 – Gestão para sucesso sustentável em uma organização. Sistemas de Gestão	Não busca caráter de certificação, mas sim de aperfeiçoamento das atividades contratuais entre clientes e as organizações.
ISO 19011 – Diretrizes para auditorias de sistemas de gestão da qualidade e/ou ambiental	Diretrizes para a verificação e inspeção dos objetivos dos sistemas e suas capacidades de cumprimento.
ISO 10003 – Guia para verificação de satisfação dos consumidores.	Diretrizes para planejamento, projeto, desenvolvimento e operações em cima das reclamações que não foram solucionadas.
ISO 10005 - Sistemas de gestão da qualidade – Diretrizes para planos de qualidade.	Diretrizes para fornecer boas práticas na análise, aceitação e revisão de planos de qualidade.
ISO 10006 – Sistemas de gestão da qualidade – Diretrizes para a gestão da qualidade em empreendimentos	Diretrizes para empreendimentos que possuem sistemas inconsistentes e que impossibilitam a realização de atividades com segurança.
ISO 10007 – Sistemas de gestão da qualidade – Diretrizes para a gestão de configuração	Diretrizes para a gestão de configuração em organizações.
ISO 10012 – Sistemas de gestão de medição – Requisitos para medição e equipamentos de medição	Define os requisitos gerais para a gestão do processo de medição e metrologia de equipamentos.
ISO/TR 10013 – Diretrizes para a documentação de sistemas de gestão de qualidade	Relatório técnico que busca instruir e capacitar as organizações no intuito de proverem uma

Comment [j183]: se são varias, sao diferentes !

percepção de benefícios financeiros e econômicos	conceitos de gestão de qualidade da ISO 9000
ISO 10015 – Gestão da qualidade – Diretrizes para treinamento	Diretrizes que orientam as organizações a elaborarem planos de treinamento com ênfase no ganho de desempenho e melhoria contínua de seus colaboradores
ISO/TR 10017 – Guia sobre técnicas estatísticas para a ABNT ISO 9001:2000	Diretrizes para a seleção de técnicas estatísticas para implantação da norma ISO 9001 nas organizações.
ISO 10019 – Diretrizes para a seleção de consultores de sistemas de gestão da qualidade e usos de seus serviços	Diretrizes para auxiliar a organização na seleção de consultores para o sistema de gestão de qualidade.
ISO/TS 16949 – Sistemas de gestão de qualidade – Requisitos particulares para a aplicação da ABNT NBR ISO 9001:2000 para organizações de produção automotiva e peças de reposição pertinentes	Direcionada para organizações de montagem e reposição de peças automotivas ou para a indústria automobilística.

O conjunto deste acervo de normas não se formou em pouco tempo e de maneira simplificada. Vários conceitos foram aos poucos sendo engajados neste quadro ao longo de vários anos com o intuito de identificar melhores práticas e suas características para serem aplicadas corretamente.

7.3.1. Gestão para qualidade nas organizações

A história de padronização para a série ISO 9000 surgiu no final da década de 80 com o governo britânico em 1987, através da extinta premissa inglesa *British Standard 5750* (BS5750). A ISO normatizou um conjunto de conceitos sobre produção e manufatura descendentes da Revolução Industrial que servem até hoje como base para guiar organizações no intuito de propor a implantação de um Sistema de Gestão para Qualidade (SGQ) [MARSHAL JUNIOR et al. 2008].

A prioridade principal especificada nas primeiras versões da norma (ISO 9000:1987) objetivava focar única e exclusivamente conceitos que relacionassem o conhecimento do adjetivo “qualidade”. “Não bastava apenas produzir e não saber melhorar”, mas sim ordenar a produção, efetuar vistorias, analisar deficiências, conter erros, desenvolver possíveis mudanças e saber aplicar a melhor alternativa possível para prover a solução e a evolução.

A primeira versão ISO 9000:1987 subdividia-se em modelos para qualidade, classificados da seguinte forma [MATOS 2009]:

- **ISO 9001** : Modelo de garantia para qualidade de projeto, desenvolvimento, produção, montagem e fornecedores aplicando-se à organizações cujas atividades eram voltadas para criação de novos produtos.
- **ISO 9002**: Modelo de garantia para qualidade na produção, montagem e prestação de serviços adequando-se da mesma documentação da ISO 9001, mas não com foco para a criação de novos produtos.
- **ISO 9003**: Modelo de garantia para qualidade na inspeção final e nos testes com foco apenas para o produto final sem nenhuma preocupação na forma como ele foi produzido.

Comment [j184]: prolix, reescrever

Comment [j185]: Normas ISO para Qualidade de Processos de Software

- **ISO 9004:** Guia de orientações conceituais sobre as definições de qualidade para sistemas e elementos essenciais para a elaboração de um Sistema de Gestão de Qualidade.

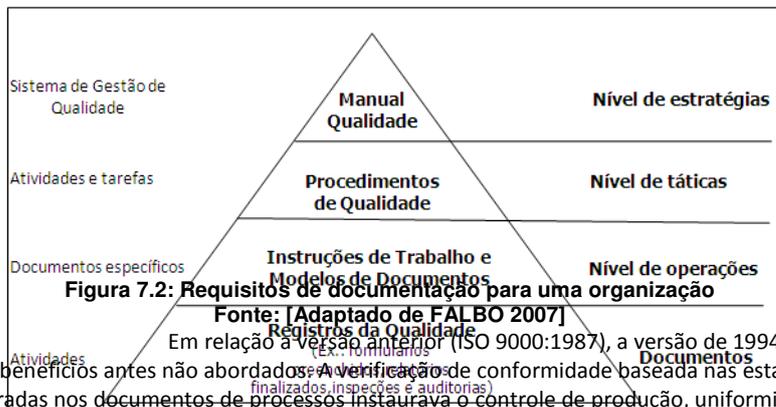
Por se tratar de especificações muito fráguas e com perspectivas baixas de negócios, o uso desta norma não rendeu os resultados esperados por muitas organizações e pela própria ISO. Seu entendimento era complicado com manuais burocráticos que restringiam algumas empresas de se adequarem a sua forma de utilização deixando muito a desejar.

Posteriormente atualizada, a ISO 9000:1994 abordava os termos técnicos para manter a garantia de qualidade contínua com a manutenção voltada para processos. A Eutech (2009) afirma que a norma não exigia que as empresas propusessem objetivos adotando ações que visassem à melhoria da qualidade, mas despertava a objeção de que as organizações proovessem documentações confiáveis para viabilizar um controle mais qualitativo e quantitativo de seus projetos e produtos: *“Document what you do, do what you document, and be prepared to prove it”* (Figura 7.2).

Comment [j186]: Normas ISO para Qualidade de Processos de Software ?

Comment [j187]: Normas ISO para Qualidade de Processos de Software ?

Comment [j188]: PROCESSOS DE SOFTWARE?



Fonte: [Adaptado de FALBO 2007]
 Em relação à versão anterior (ISO 9000:1987), a versão de 1994 trouxe benefícios antes não abordados: A verificação de conformidade baseada nas estatísticas registradas nos documentos de processos instaurava o controle de produção, uniformização e prevenção buscando amenizar as reincidências de não cumprimento das metas estabelecidas no plano de projeto de um produto ou serviço [MUTAFELIJA e STROMBERG 2003]. A conscientização da alta direção, além de vários fatores estratégicos que discernem melhorias no tratamento de redundâncias, como por exemplo, rastreabilidade, provisionamento, controle de concepção e métricas por função dinamizavam o processo de avaliação de uma empresa ajudando-a na implantação e manutenção de um Sistema de Gestão para Qualidade.

Mesmo utilizando uma temática mais comercial para inserção de qualidade, a ISO 9000:1994 deixou muito a desejar em vários pontos de sua documentação. Os termos e a simbologia adotados eram complexos muitas vezes inviabilizando sua aplicação em determinadas empresas. A visão restrita apenas para processos não concebia um guia para a resolução de problemas de desempenho nas organizações, o que muitas vezes trazia dificuldades ainda maiores de implantá-la e principalmente conseguir mantê-la sem danificar ou prejudicar o fluxo de funcionamento da empresa.

A solução para este problema se deu na segunda revisão realizada em 2000. A ISO 9000:2000 mudou completamente o pensamento e a ideologia de dedicação exclusiva para processos abordando os principais fundamentos e um vocabulário mais objetivo baseado nas experiências coletadas para implantação de qualidade nas organizações [SIMÕES et al. 2003].

A nova atualização da série ISO 9000 trouxe consigo uma base mais consistente tratando assuntos mais atuais e as reais necessidades de melhoria que precisam adequadas para

- **Foco no cliente:** O pilar prioritário da norma estabelece em focalizações constantes que busquem as necessidades e expectativas do cliente. A competitividade no mercado existe e enfoca em torno dos clientes, que são os principais responsáveis pelo crescimento ou fracasso das organizações.
- **Liderança:** Os líderes devem ser conscientes que precisam despertar os propósitos e as metas para a busca de resultados na organização. A liderança que a norma instaura releva para os líderes saberem implantar valores como dedicação, determinação e empenho dos envolvidos.
- **Envolvimento das pessoas:** Os colaboradores precisam estar conscientes que são as peças-chaves para o progresso da organização. Suas habilidades precisam ser lapidadas e aperfeiçoadas para que os resultados apareçam com suas aplicações.
- **Abordagem do processo:** A imposição de atividades definidas e com práticas bem apresentadas formaliza os passos certos para desenvolver processos executáveis em organizações.
- **Abordagem de sistemas para gestão:** Planejar, desenvolver e tentar aplicar uma temática de “sistematização” das atividades da empresa, neste caso, os processos, unindo-os para possibilitar análises, manutenções e principalmente melhorias nas interações existentes entre eles.
- **Melhorias contínuas:** Não basta apenas produzir. A melhoria e evolução são fatores que emanam confiança e segurança nos produtos e serviços diferenciando-os entre qualitativos ou não.
- **Abordagem factual para tomada de decisões:** A análise dos dados, principalmente das documentações, relatórios, etc., ditam decisões coesas e seguras habilitando a organização para obtenção de perspectivas favoráveis e resultados positivos.
- **Relacionamento com fornecedores visando benefícios mútuos:** Uma organização é um conjunto de organizações. Boas relações com os fornecedores são indispensáveis para o fluxo de funcionamento da organização como um todo. Investimentos entre ambos agregam valores tais como confiança, segurança e estabilidade.
Com a inserção de novas metodologias, para uma nova visão de processos e pessoas na ISO 9000:2000, a estrutura da família também sofreu algumas modificações e atualizações. As principais, incluindo as normas descentes da ISO 9000 segundo a DQS (2009) foram:
 - **ISO 9000:** Sistemas de Gestão de Qualidade – Conceitos e Terminologia. Substitui a ISO 8402 e 9000-1.
 - **ISO 9001:** Sistemas de Gestão de Qualidade – Requisitos. Substitui ISO 9001/2/3
 - **ISO 9004:** Sistemas de Gestão de Qualidade – Guia para melhoria de desempenho. Substitui a ISO 9004-1.
 As mudanças efetuadas na ISO 9000:2000, no âmbito de Gestão de Qualidade, são as principais responsáveis por manterem uma base sólida para a formulação e desenvolvimento das certificações ISO 9001:2000 e ISO 9001:2008. Muitas empresas e pessoas confundem ISO 9000 e ISO 9001 achando que as mesmas são certificações diferentes ou estágios de melhorias baseadas em avanços na obtenção de qualidade. A ISO 9000:2000 não possui vínculo certificador, mas apenas conceituador, legislando nas certificações ISO 9001 um padrão de conceitos, vocabulários, termos e requisitos mínimos para que as avaliações instauradas nas

A mais recente atualização da série ISO 9000 aconteceu em 2005. Uma revisão de conceitos foi realizada com intuito de prover o entendimento mútuo da terminologia utilizada na versão anterior facilitando o comum acordo entre fornecedores, clientes, órgãos reguladores e certificadores. A Target (2005) afirma que algumas técnicas que figuravam nas normas anteriores foram remodeladas na versão ISO 9000:2005. Termos antes não evidenciados como: Perícia técnica, auditor técnico, exigência de metas, competências, pro – atividade com liderança, análise de contrato, controle de artefatos, revisor de contas, equipe de apoio, planta de apoio, reuniões de emergências, etc.

No Brasil a tradução e regulamentação da ISO 9000 ficam a cargo da ABNT. Sob o formato de Norma do Brasil (NBR) ISO 9000 várias empresas brasileiras buscam adaptar-se a exigências impostas pelos guias de referência da norma desenvolvendo Sistemas de Gestão de Qualidade com foco principalmente para certificação ISO 9001. Assim como várias outras normas descenderam da série ISO 9000, uma das principais, a ISO 9001, exerce forte influência para normatizar organizações seguindo um longo e rigoroso guia de requisitos para obtenção de qualidade nos seus processos e produtos.

7.3.4. Norma ISO 9001

Com o lançamento da ISO 9000, várias organizações despertaram a temática de que precisavam impor, e principalmente manter, padrões de qualidade em seu funcionamento, seja nos processos, ou mesmo nas pessoas que colaboram para o funcionamento das mesmas. O pensamento com uma melhor visão e ambição para o mercado dispõe da realização de investimentos que prestem alternativas viáveis para o crescimento e melhoramento das atividades.

Mello et al. (2009) descreve que as normas para sistemas de gestão, principalmente a ISO 9001, fornecem modelos básicos para que as organizações preparem e operem seus fluxos de funcionamento seguramente fortificados. O autor ainda cita que:

“As grandes organizações, ou aquelas com processos complexos, poderiam não funcionar bem sem um sistema de gestão, apesar de ele poder ter sido chamado por algum outro nome.”

A norma ISO 9001 foi instituída com esse propósito. Descrever os requisitos para possibilitar a implantação e administração de um modelo para garantia de qualidade para produtos e serviços através de um Sistema de Gestão de Qualidade. Como estratégia de negócios para apresentar uma base sólida de segurança e qualidade nas empresas, esta norma condiz um fator de certificação através de auditorias, inspeções, dentre outras atividades que classifiquem e garantam boa procedência para verificação e validação de processos e serviços conforme as terminologias e vocabulários apresentados pela ISO na versão 9000.

7.3.5. Certificação ISO 9001

O termo certificação inflige características bem perplexas em seu significado. Ferreira

diferentes considerando os padrões que precisam ser mantidos no desenvolvimento de suas certificações.

A ISO 9001:1994 surgiu como a primeira versão em caráter avaliativo para a certificação de Sistemas de Gestão de Qualidade. Baseada em vinte elementos-chaves³ para facilitar a administração das organizações, esta certificação adotou políticas definidas principalmente para gerência de processos e produtos para fábricas em vários níveis de produção [MUTAFELIJA e STROMBERG, 2003]. Melloti et al. 2007 descreve que esta norma possuía uma visão desmembrada de negócios para organizações. A adoção de seus requisitos era instaurada nos processos para a formação de um sistema de qualidade, porém de forma paralela as relações existentes entre as organizações e os fornecedores, muitas vezes dificultando a exclusão de problemas que influenciavam em todo o sistema de gestão adotado.

O uso da versão de 1994 estava subsidiado a elaboração de vários documentos diferentes. Como a primeira certificação abordava um conjunto de várias normas ao mesmo tempo, as organizações precisavam elaborar planos de gestão que adotassem medidas seguras para possibilitar o acompanhamento das práticas e técnicas sugeridas nos vocabulários da ISO 9000:1994. Os requisitos da ISO 9001:1994 muitas vezes se faziam diferentes da estrutura real de muitas organizações obrigando-as a remodelarem suas atividades e tarefas para tentarem se engajar ao modelo de requisitos solicitado pela norma para a termos de certificação. Para apagar a imagem de inadequação da ISO, em termos de avaliação para a época, em 2000 a certificação ganhou uma nova revisão instaurando novas perspectivas de mercado para processos e produtos, viabilizando adotá-la para obtenção da certificação.

O efeito de mudanças para a época não gerou resultados tão satisfatórios quanto esperados pela ISO. A complexidade aliada a uma difícil tradução e implantação sintetizou mudanças desagradáveis para adequação, e principalmente, a administração de gestão de qualidade em propor novas avaliações e melhorias nas organizações e seus Sistemas de Gestão para Qualidade.

A ISO 9001:2000 foi lançada com o objetivo de incluir o cliente como ponto-chave nos processos. Assim como a versão anterior, a atualização de 2000 possui descrições genéricas, possibilitando as organizações a implantarem seus requisitos em seus Sistemas de Gestão para Qualidade independente de porte, produtos ou serviços fornecidos [SPINOLA 2005]. A quantidade de elementos-chaves em relação à versão de 1994 foi reduzida deixando a norma mais consistente para propor um entendimento mútuo entre os fornecedores, as organizações e os clientes. Spinola (2005) destaca alguns dos elementos-chaves fundamentais (Tabela 7.2) da ISO 9001:2000 tais como:

Tabela 7.2: Elementos-Chave da ISO 9001:2000

Fonte: [SPINOLA 2005, p. 29]

Elementos Chaves
Satisfação do cliente
Abordagem de processo
Indicadores de desempenho
Melhoria contínua

Observa-se na Tabela 1 que a norma engloba quatro principais referências para gestão.

O cliente está acima de tudo, em uma visão de que as metas de qualidade norteiam sua satisfação para com a organização. Para Mello et al. (2009), as organizações devem desenvolver práticas e técnicas com fluxos de funcionamento aplicáveis para imposição de melhorias qualitativas em função dos clientes usando-se da seguinte premissa:

Comment [j189]: Tabela de uma coluna?

“Entender todas as necessidades e expectativas do cliente relativas aos produtos, prazo de entrega, preço, confiabilidade, etc.”

Ainda segundo o autor, outros pontos fortes tais como a adoção de uma boa comunicação entre as organizações e os clientes, além de medições de satisfação dos clientes e a atuação das organizações sobre estes resultados, devem despertar perspectivas mais previsíveis e um ganho de mercado cada vez mais seguro.

A abordagem de processo imposta pela ISO 9001:2000 condiz à descrição de maneira clara e concisa das principais atividades e tarefas para a descrição e avaliação dos processos. A sistemática de independência entre eles deve ser implantada com o gerenciamento dos artefatos de entrada e saída, estipulando o desacoplamento da estrutura da organização para facilitar a inserção, alteração e remoção dos processos e pessoas que assumem cargos nas mesmas, sem deixar o sistema fragilizado para influenciar nos resultados.

O desempenho e o ganho dele também são primordiais para obtenção de qualidade. O pensamento de impor melhorias não apenas nos processos, mas sim com investimentos nas pessoas que os executam, dinamiza o crescimento dos indicadores para bons resultados através de estratégias qualitativas que buscam unificar as metas e descrições dos processos adjuntos as habilidades exercidas pelos colaboradores.

Para o desenvolvimento da melhoria contínua, a certificação estimula a utilização do ciclo *Plan-Do-Check-Act* (PDCA). Planejar, checar, verificar e agir sintetizam aspectos de aprofundamento nas características dos elementos chaves citados anteriormente, como também para um próprio melhor conhecimento dos processos, possibilitando a aplicação de modelos de melhorias para processos, como por exemplo, o uso de *frameworks* e ferramentas como o CMMI, o MPS. BR, dentre outras [MUTAFELIJA e STROMBERG 2003].

7.3.6. ISO 9001:2008

A nova e recém formulada certificação para sistemas de gestão de qualidade é a ISO 9001:2008.

A certificação enfoca basicamente o mesmo contexto de sua anterior, adicionando apenas algumas mudanças significativas para a melhoria de entendimento e implantação dos requisitos nos sistemas de gestão de qualidade adotados [MELLO et al. 2009].

O prefácio da norma foi reestruturado em várias seções e subseções ao longo de sua documentação. Alguns pontos retrógrados, ainda referenciados da extinta versão de 1994 foram banidos, e em boa parte da descrição dos itens foram inclusas referências para a avaliação e concepção de produtos e serviços envolvendo principalmente aspectos gerenciais para processos. A estrutura composta de oito tópicos desmembra práticas que, aliadas aos guias técnicos da ISO 9004, complementam um conjunto de diretrizes para a busca de melhorias contínuas, e principalmente ganhos de desempenho constante, explicitando a qualidade inerente de boas técnicas e metodologias que determinam o sucesso nos projetos instituídos pelas organizações.

A versão em uso da NBR ISO 9001:2008 no Brasil é a segunda, publicada em novembro de 2008 e validada em dezembro do mesmo ano [ABNT 2008]. Adequada do modelo original elaborado pelo comitê ISO/TC 176⁴, esta certificação possui no escopo termos definidos como “generalidades” que capacitam os consultores a estipularem planos de análises para processos de acordo com os requisitos e seus fatores de implantação. A [ABNT 2008a] descreve o sumário da ISO 9001:2008 na seguinte abrangência de assuntos de gestão para qualidade:

- Introdução: Possui características correlatas as generalidades superficiais da norma destacando o conceito da abordagem de processo e cliente, com o PDCA, a relação da certificação com a ISO 9004 e a compatibilidade com outros sistemas de gestão⁵
- 1 - Escopo: Neste item são apresentadas generalidades de aplicação segundo os vocabulários da NBR ISO 9000:2005.
- 2 – Referência normativa: Complementando o tópico 1, destacam-se os termos e fundamentos da NBR ISO 9000:2005: *Sistemas de Gestão da Qualidade*
- 3 – Termos e definições: Padroniza aspectos e palavras como “produto” e “serviço” para que não sejam confundidos durante a abordagem.
- 4 – Sistemas de Gestão da Qualidade: Os primeiros requisitos gerais da certificação descrevem “práticas base”. Determinar os processos, suas interações existentes, além de monitorar e implementar metodologias para a elaboração de estratégias para os mesmos, influencia o desenvolvimento de manuais para qualidade e o controle de documentos e registros, que deliberam segurança e garantias para a realização das atividades e tarefas dentro de um sistema de gestão.
- 5 – Responsabilidade da direção: Destina-se a conscientização para com os líderes das organizações. A alta direção deve definir estratégias para serem executadas nos níveis táticos e operacionais (Ver Figura x). A norma cita que um fator diferencial para que se obtenha isto é a especialização da comunicação entre as categorias que formam o sistema, além da análise crítica de realimentação de mudanças que suscitam progressos baseados em ações de acompanhamento com contenções e prevenções.
- 6 – Gestão de recursos: Um ponto importante, na implantação da qualidade envolvendo clientes, organizações e fornecedores, diz respeito à administração dos recursos. A provisão, a qualificação e o melhoramento de perspectivas com recursos humanos, com uma mão de obra de boa procedência, por exemplo, idealizam a valorização de investimentos em treinamentos, infraestrutura física e matérias primas adequadas que insiram ganhos de consciência para todos os envolvidos com o intuito de que a meta de competência estabelecida seja alcançada.
- 7 – Realização do produto: Todos os processos sucumbem em produtos que precisam estar de acordo com as reais necessidades dos clientes. Para isto, a organização deve prover planejamentos baseados em pesquisas e análises constantes de entradas e saídas de projetos e desenvolvimento, visando verificar e validar mudanças demandadas que satisfaçam a propriedade dos clientes e a preservação do produto. A realização do produto, segundo a ISO 9001:2008, deve constar de um controle de equipamento e o monitoramento de medição.

⁵ Segundo a norma, a introdução da ISO 9001:2008, apresenta uma abordagem de processo e cliente, com o PDCA, a relação da certificação com a ISO 9004 e a compatibilidade com outros sistemas de gestão.

- 8 – Medição, análise e melhoria: As medições nas organizações para o ganho de qualidade são inevitáveis. O acompanhamento constante e sua avaliação momentânea são frutos de auditorias rígidas e detalhistas nos processos e produtos. A certificação exige que os projetos sejam executados baseados em dados concretos e seus resultados conforme o andamento das atividades que angariam as melhorias impostas de acordo com a evolução do produto ou do serviço.

A norma consta ainda de dois anexos (A e B), ambos em caráter informativo, e uma bibliografia proveniente de outras normas da série 9000. O primeiro anexo expõe a correspondência da ABNT NBR ISO 9001:2008 com a ABNT NBR ISO 14001:2004 apresentando diretrizes de implantação de sistemas de gestão para qualidade envolvendo aspectos ambientais. No segundo são idealizadas as principais diferenças na atualização da ABNT NBR ISO 9001:2000 com a ABNT NBR ISO 9001:2008 referenciando e idealizando o que foi adicionando, alterado e removido, para facilitar a atualização dos sistemas para as organizações que possuem a certificação ISO 9001:2000 [ABNT 2008].

O processo de implantação da certificação é burocrático e extenso. De início a organização deve estabelecer um formato de funcionamento denominado *unidade de negócio*, que se compõe de pessoas, informações e responsabilidades para que todos unifiquem uma sociedade. Formada a unidade e sua regulamentação, a organização deve instituir os principais elementos básicos, tais como missão, visão, fornecedores, insumos, macro (ou sub) processos, produtos e indispensavelmente o cliente alvo [MELLO et al. 2009].

Outro ponto importante é a adoção de uma política e objetivos da qualidade. A alta direção impõe um plano de metas que devem ser analisadas pelas gerências e posteriormente realizadas pelos demais colaboradores. Os objetivos são mensurados em números em uma escala de análise nos pontos estratégicos e nas correspondências de suas aplicações. Cumprimento de prazos, redução de erros e contenções de gastos, dentre outros detalhes ínfimos que fazem a diferença, idealizam o atendimento das necessidades explícitas e implícitas dos clientes, fornecedores e da organização durante o processo de implantação deduzindo-se então que a qualidade pode ser aplicada sem nenhuma restrição.

O mapeamento e a descrição dos processos também se integram nos requisitos para a obtenção da certificação. Metodologias como o *Business Modeling Process*⁶ e a utilização do PDCA facilitam a abordagem dos processos delineando a padronização e identificação de procedimentos, instruções e características que controlam as atividades e tarefas básicas para a elaboração de um plano de sistematização de qualidade aplicável, tornando-se este, padrão para a organização e como modelo de gestão para ser adotado.

Por fim, a solicitação de um órgão consultor para a realização da auditoria nos padrões de requisitos da norma. A entidade credenciada pela associação nacional deve elaborar um calendário de visitas sequenciais para a realização de análises, verificações, documentações e todo um processo de compatibilização da organização para que esta se enquadre dentro dos padrões da ISO para poder classificá-la como qualitativa e concedê-la o certificado ISO 9001 de qualidade.

7.3.7. ISO/IEC 90003

A abrangência genérica para a sistematização da qualidade em organizações inserida pelas certificações ISO traz conceitos que muitas vezes não identificam as práticas específicas para a gestão de processos ou produção de software, para serem implantadas em projetos. A ISO, em parceria com a IEC, desenvolveu um guia de referência que buscasse complementar a

Comment [j190]: Isso sim é software!

aplicação da certificação ISO 9001 com o propósito de normatizar e qualificar a gestão da qualidade baseado nas chamadas *Fábricas de Software*⁷.

A norma ISO/IEC 90003 é uma atualização da extinta terceira parte da ISO 9000 (9000-3). O propósito apresentado por esta norma em forma de guia de referência, não foi o de certificar, mas sim, auxiliar as organizações na aquisição, fornecimento, desenvolvimento, operação, e manutenção de software, identificando as alternativas para fortalecer os processos adotados pela instituição, independente da tecnologia utilizada, dos modelos de ciclo de vida, do processo de desenvolvimento utilizado, e principalmente da estrutura atual em que se encontra a organização [SPINOLA 2005].

Para Cortês (2008), a descrição e a terminologia da ISO/IEC 90003 estão relacionadas de acordo com os requisitos descritos na ISO 9001. Para cada requisito, e o conjunto de componentes que os formam, são realizadas interpretações que adéquam as características genéricas de gestão de qualidade para sistemas, ao contexto do desenvolvimento de produtos de software, através de orientações operacionais e técnicas provenientes da certificação, tais como *shall* (deve fazer), *should* (poderiam ou convém que) e *may* (podem fazer), melhorando o fluxo de funcionamento do ciclo de vida dos processos com práticas seguras que geram confiabilidade nas ações e decisões evidenciadas pela alta direção sendo executadas pelos colaboradores dos níveis táticos e técnicos.

A estrutura da norma é basicamente caracterizada por um conjunto de atividades essenciais que precisam ser adotadas durante a produção de um software. Marinho (2007), afirma que a ISO/IEC 90003 estabelece detalhadamente as responsabilidades e ações que devem ser tomadas em relação ao ganho qualitativo organizacional eminente das práticas provenientes da adoção de sistemas de gestão compatíveis com as estruturas dos projetos em andamento. As atividades classificadas como *Atividades de ciclo de vida* e *Atividades de suporte*, são descritas da seguinte forma [MARINHO 2007]:

- **Atividades de ciclo de vida:** Determinam práticas munidas de ações preventivas e corretivas que devem ser inseridas durante os ciclos de vida provenientes de processos durante o desenvolvimento de um do software. As atividades de ciclo de vida da ISO/IEC 90003, segundo Marinho (2007), são:
 - Análise crítica de contrato: Informa e conforma os itens mínimos que devem compor um contrato de software evidenciando principalmente a segurança das informações e os aspectos de autoria técnica e de trabalho.
 - Especificação de requisitos do comprador: Descreve os aspectos que ditam as necessidades implícitas para efetivar melhorias relacionadas aos requisitos estabelecidos pelos compradores. Alguns desses aspectos são confiabilidade, desempenho e segurança.
 - Planejamento do desenvolvimento: Desperta e atribui as atividades para melhorar a administração e qualificar o desenvolvimento do software, com a elaboração de cronogramas, definição das fases, planos de testes, dentre outras técnicas que facilitem o andamento e a conclusão do desenvolvimento durante o projeto.

Comment [j191]: Ã?

- Planejamento da qualidade: Identifica algumas práticas para instituir melhorias contínuas durante o projeto do software evitando assim retrabalhos constantes, reprocessamentos desnecessários, manutenções seqüenciais e menos ciclos de treinamentos, além de tornar mais maduros, os processos quantificados com o desenvolvimento e catalogação de documentos de artefatos.
- Projeto e implementação: Idealiza atividades para que projeto forneça resultados consistentes e satisfatórios baseados nas experiências dos profissionais que os executam. O uso de metodologias e processos de desenvolvimento é abordado para alertar a busca para garantir um desacoplamento do software para que o mesmo discorra totais condições de receber manutenções prévias sem afetar a estrutura do sistema como um todo.
- Teste e validação: Externa a necessidade para a elaboração de planos de testes do software e a homologação dos resultados em vários níveis. Um plano de testes deve possuir tópicos de análise e execução baseados no ambiente, na documentação, nos casos de testes e principalmente na quantificação e análise de comparação efetivada com os dados.
- Aceitação: Adentra uma série de atividades para viabilizar a utilização de testes de aceitação, procedimentos para avaliação, ambiente e recursos de hardware e software, além de constantes diálogos entre o comprador e o desenvolvedor para identificar os fatores de conformidade.
- Reprodução, expedição e instalação: Esta parte da norma abrange as regras que guiam a administração do número de cópias, tipos de meio físico utilizado, licenças e direitos autorais. Boa parte dos requisitos analisa as obrigações que precisam ser impostas através da elaboração de direitos e deveres utilizados pelos compradores e desenvolvedores relativos à instalação dos sistemas.
- Manutenção: Identifica e analisa a manutenção como um fator indispensável para a implantação da qualidade. Todo e qualquer produto de software precisa constar de correções em intervalos de tempos definidos para capacitar sua estrutura a receber melhorias posteriores conforme as necessidades do comprador.
- **Atividades de suporte:** Determinam novos itens que devem ser implementados pelo fornecedor do software a medida que o sistema fora sendo adequado a organização compradora. Estas atividades não compreendem nenhuma parte ou fase do ciclo de vida do software, mas sim atividades de apoio que auxiliem a gestão e administração do software para com a estrutura da organização.
- Gestão de configuração: Estabelece as descrições necessárias para prover a rastreabilidade ideal de modo que se torne possível identificar as versões do software, as atualizações, as alterações, e todo o contexto que abrange as atividades e tarefas de administração preventiva e corretiva do software.

- Controle de documentos: Estabelece o controle que deve ser feito pelo fornecedor de todos os procedimentos realizados antes, durante e após o projeto relatando a seqüência dos processos e seus resultados alcançados. Nesta parte da norma é idealizada a inserção de qualidade durante todo o ciclo de vida de produção do software através de descrições que precisam ser feitas e apresentadas aos clientes para a verificação de conformidade.
- Registro de qualidade: A norma aconselha e descreve que o fornecedor deve constar de artifícios que lhe concedam formas de coletar, analisar, manter e comparar registros de ganhos quantitativos e qualitativos de forma que sejam constantemente recuperáveis.
- Medição: Institui e estimula a praticado uso de métricas de software para possibilitar medições nos processos e produtos desde o planejamento até o fornecimento de manutenção e treinamento. A norma cita alguns exemplos tais como: Números de falhas por uso efetivo, tempo médio de reparo do problema e tempo médio entre duas falhas consecutivas.
- Regras, práticas e convenções: A norma cita que cada fornecedor adjunto ao comprador deve definir regras e convenções de termos comuns entre ambos para facilitar o contato e a implantação da qualidade de acordo com a metodologia, técnica ou *framework* adotado para melhoria de processos e produtos.
- Ferramentas e técnicas: A norma condiz ao fornecedor prover todos os recursos cabíveis para angariar com as descrições correntes da norma no intuito de cumprir seus requisitos de implantação de qualidade nas Fábricas de Software.
- Aquisição: Descreve os fatores de comprometimento que devem existir entre o fornecedor e o comprador para proporcionar a conformidade dos requisitos apresentados no início e sua veracidade suprida no produto final.
- Produto para ser incluído no software: A norma cita que a qualidade deve ser distribuída em várias partes de um projeto ou processos, não sendo analisada um âmbito macro.
- Treinamento: Destaca a importância em que o fornecedor deve proporcionar a capacitação para os utilizadores do software após sua implantação ou em intervalos de tempo comunicáveis melhorando assim a usabilidade e o desempenho interno para a busca de qualidade contínua.

Como pode ser observado nos tópicos acima, as atividades de ciclo de vida descritas pela a ISO/IEC 90003 relatam aspectos comuns a muitos projetos de software. Os requisitos sintetizados visando a integração de processos de software aos sistemas de gestão das organizações despertam um conjunto de práticas da ISO 9001 tratando-se mais especificamente da relação da fábrica com o próprio cliente, visto que a qualidade para o software é alcançada em virtude do atendimento por completo das necessidades implícitas e

Outro ponto de destaque abordado pela a ISO/IEC 90003 é o complemento de apoio organizacional. As atividades de suporte apresentadas pela norma inserem um contexto que evidencia aspectos que emanam qualidade nos processos similarmente a própria certificação, apenas mudando a terminologia e os vocabulários caracterizando-os mais especificamente para software.

A norma ISO/IEC 90003 assim como citada anteriormente é uma complementação da ISO 9001. Com o lançamento da versão de 2008, o guia técnico mais atual encontra-se ainda em fase de atualização para adequar-se as mudanças realizadas e propor melhorias conforme os requisitos apresentados acima. Estima-se que a próxima versão desta norma não seja tão afetada o quanto a ISO 9001:2008 foi em relação a sua versão anterior, simplificando assim, os trabalhos de atualização e migração de processos de software às suas descrições.

7.4. ISO/IEC 12207

Comment [j192]: Isso é software!

As normas que certificam Sistemas para Gestão da Qualidade especificam fatores e requisitos que precisam ser cumpridos pelos colaboradores da organização à medida que os processos são executados e os produtos são desenvolvidos. Se por um lado, avaliar os critérios dos processos, suas características, diretrizes e restrições, é importante para discernir e quantificar os conceitos da qualidade com as certificações, por outro lado, organizar os processos e saber classificá-los, atribuindo-lhes atividades e descrevendo suas tarefas, torna-se indispensável para que se proponha a implantação da temática de Qualidade de Software com normas ISO para processos em organizações.

A adoção da ISO /IEC 12207 possibilita a utilização desse contexto mencionado. Estimando-se um controle seguro para o ciclo de desenvolvimento e o acompanhamento de todos os processos que formam um dito projeto, o foco de abordagem desta norma de cunho não certificador, abrange apenas a definição de uma terminologia simples para Fábricas de Software, instruindo-as para adentrar de características qualitativas e organizacionais que precisam ser adequadas em suas fases, desde as iniciais até as finais, para contribuir na identificação e classificação de seus processos e definir plausivelmente suas características essenciais [MACHADO 2006].

Os objetivos principais descritos pela ABNT (1998) descrevem uma norma de fácil entendimento e utilização. Diferentemente da série ISO 9000, a ISO/IEC 12207 não impõe um padrão criterioso e detalhista para avaliação e mensuração qualitativa de processos, mas sim, institui que as organizações possuam total liberdade para identificar, executar e administrar as atividades e tarefas de forma integrada, independente da maneira como os processos possam estar interligados, com o proposto de que os mesmos possam comportar mudanças que não inflijam ou alterem funções e características dos demais processos.

7.4.1 Estrutura da norma: Processos de ciclo de vida

Diferentemente da norma ISO/IEC 15504, que será abordada posteriormente neste capítulo, a ISO/IEC 12207 possui uma estrutura mais simplificada e objetiva. Publicada em 1º de agosto de 1995, a norma oferece outra perspectiva descrita por [KOSCIANSKI e SOARES 2007] da seguinte maneira:

“Não define objetivos, níveis de maturidade organizacional ou de capacidade de processo, mas sim dispõe de uma estrutura mínima para que a organização defina seus

Segundo a ABNT (1998), os processos desta norma formam um conjunto abrangente. Dependendo do seu objetivo, a organização pode selecionar a quantidade e os processos específicos que lhe convenham mais viáveis para o projeto em questão. A ISO/IEC 12207 é, portanto, projetada para ser adaptada para uma Fábrica de Software que busque em dispor das etapas mínimas que devem ser implantadas no gerenciamento do ciclo de vida de um sistema de informação, independente da estrutura funcional adotada, para propiciar o conhecimento máximo para cada processo no intuito de adequar um conceito de execução gerenciável que deve ser adotado pela organização.

Para cada processo são atribuídas atividades e tarefas. Como as organizações variam seus ciclos de vida de acordo com a complexidade dos projetos, a norma medrou um esboço simples e bastante eficaz que sintetiza nos engenheiros a ideia de elaborar mais detalhadamente os aspectos que caracterizam a formação de cada um. A estrutura básica que formar os processos (Figura 7.3) segundo a norma ISO/IEC 12207 é a seguinte:

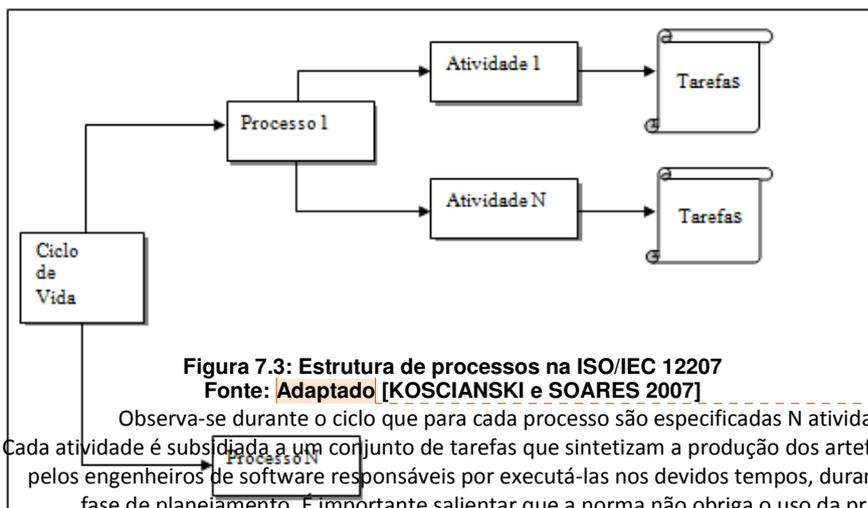


Figura 7.3: Estrutura de processos na ISO/IEC 12207
Fonte: Adaptado [KOSCIANSKI e SOARES 2007]

Observa-se durante o ciclo que para cada processo são especificadas N atividades. Cada atividade é subsidiada a um conjunto de tarefas que sintetizam a produção dos artefatos pelos engenheiros de software responsáveis por executá-las nos devidos tempos, durante a fase de planejamento. É importante salientar que a norma não obriga o uso da prática sugerida na Figura x, porém adverte que a organização deve adotar o meio mais viável para descrever as principais características que formam os processos de ciclo de vida adotados para o projeto de sistema.

Os processos que compõem o chamado “ciclo de vida”, descrito pela ISO/IEC 12207 estão padronizados através de um contingente de ordem de execução e utilização distribuídos em três categorias principais: Processos Fundamentais, Processos de Apoio e Processos Organizacionais, além de um processo auxiliar, intitulado de Processo de Adaptação [MACHADO 2006]

Comment [j193]: Centralizar descrição dos quadros

Comment [j194]: de

Comment [j195]: .

7.4.2 Processos fundamentais

Nesta categoria estão incluídos os processos básicos responsáveis pela produção de um

- **Processo de aquisição:** A aquisição ocorre quando a organização busca uma solução customizada para a fabricação do software e o atendimento imediato das necessidades do cliente. A aquisição de ferramentas pode ocorrer através de um contato com outra fábrica de software ou simplesmente com a compra de “produtos de prateleira”, como por exemplo, editores de textos e ferramentas RAD⁸ de desenvolvimento. As atividades prioritárias que descrevem este processo são o lançamento de uma proposta, o pedido de formulação de um contrato, a monitoração da relação de um fornecedor e seu cliente, além da aceitação e conclusão de todos os fatores e aspectos que idealizam e justificam o início do projeto.
- **Processo de fornecimento:** Neste processo é realizada uma proposta para a revisão e finalização do contrato. São analisados recursos genéricos para gerenciar e discutir o projeto, garantindo sua execução do início até o fim. As atividades prioritárias exercidas neste processo condizem ao planejamento, execução e controle, revisão e avaliação, e entrega e conclusão dos produtos e serviços comprados ou prestados pelo fornecedor conforme estipulado em cumprimento ao contrato efetuado.
- **Processo de desenvolvimento:** O desenvolvimento não corresponde simplesmente à codificação em si. Neste processo são relacionadas às atividades de levantamento de requisitos, análise, codificação, testes, implantação e aceitação, selecionando ferramentas e técnicas para que o produto alcance um padrão de qualidade cada vez mais superior.
- **Processo de operação:** Constitui atividades paralelas que devem ser realizadas entre a utilização do software e o suporte ao usuário. Novas atualizações, expansões, e orientação dos usuários, além da definição de execução do próprio processo para a operação do produto, constituem as principais atividades desta fase de complementação recíproca do fornecedor para com o cliente.
- **Processo de manutenção:** Contém as atividades de solução de problemas do produto em questão. O processo é executado quando são realizadas alterações de código, documentações técnicas ou contratuais, em virtude da correção de erros para impor melhorias contínuas que possibilitem a implementação da modificação, a revisão dos fatores avaliados, a aceitação das alterações, a migração de dados, ferramentas, tecnologias e operações, como também a descontinuação do software através de modificações de sistemas legados e suas devidas reparações quanto as funções e responsabilidades.

Os processos primários são os processos ditos “essenciais”. Nenhum projeto pode ser iniciado sem no mínimo realizar estes cinco procedimentos citados anteriormente. Independente da nomenclatura determinada, a organização adquire ferramentas de produção para desenvolver um software, fornecendo através de um processo de desenvolvimento, um produto que será implantado e que posteriormente possuirá manutenções. Para fortalecer a

consistência destes processos, são utilizados os processos secundários, denominados de Processos de Apoio.

7.4.3 Processos de apoio

Nesta categoria estão inclusos os processos que são executados correlacionados aos denominados processos primários. Um processo de apoio só pode ser aplicado, quando antes dele, já estiver sido iniciado um processo primário. Os processos de apoio da ISO/IEC 12207, descritos pela ABNT (1998) são:

- **Documentação:** Neste processo é estimulada a prática de desenvolver documentações para o ciclo de vida, com o intuito de validar as atividades realizadas durante o projeto, além de tornar verídica a realização do mesmo. Uma documentação bem elaborada deve conter padrões de estética, numerações de seções, tabelas e figuras, com o intuito de prover uma compreensão bem objetiva das informações de arquitetura do software para fortificar a gestão da qualidade que dever adotada pela organização e implantada nos processos durante a catalogação dos artefatos de entrada e saída produzidos.
- **Gerência de configuração:** Presume a ideia de gerenciar todos os artefatos, contabilizando suas diversas versões produzidas durante o ciclo de vida do processo. A importância deste processo está denominada na constante prática de reuso que é utilizada na concepção do produto, à medida que o projeto é realizado, além de avaliar, gerenciar e liberar as cópias do software e das documentações provenientes de uma determinada iteração concluída durante a execução do projeto.
- **Gerência da qualidade:** Neste processo são realizadas verificações para averiguar se os produtos satisfazem os requisitos e se a execução dos processos está em conformidade ao que foi planejado. A norma cita que a qualidade está diretamente relacionada à implantação correta dos requisitos para com as funcionalidades atribuídas ao produto final com a inspeção e validação dos processos e produtos, determinados por um Sistema de Gestão para Qualidade que deve ser utilizado pela Fábrica de Software, garantindo assim que os procedimentos adotados ao longo do projeto culminem em ações que posterguem a qualidade máxima alcançável focando-se principalmente a fidelização do cliente e a expansão de mercado com os investimentos realizados pelas as organizações.
- **Processo de verificação:** Neste processo são realizadas verificações de funcionalidades em cada artefato produzido durante o projeto. A meta de verificação condiz evitar os desvios de implementação que podem ser obtidos caso as atividades não estejam bem especificadas e definidas pela equipe de desenvolvimento.
- **Processo de validação:** Consiste em determinar se o produto final corresponde ao objetivo pela qual foi designado. Neste processo são realizados vários Testes de Software, como por exemplo, teste unitário e testes de estresse.
- **Processo de revisão conjunta:** Na avaliação conjunta são realizadas verificações do processo em relação aos artefatos produzidos. As revisões de gerenciamento de projeto incluem reuniões previamente marcadas ou de caráter emergencial caso se tenha

produtos de software para verificar se as alterações foram efetuadas corretamente para que não surjam problemas posteriores que comprometam os demais processos ou tornem vulneráveis os produtos que deles descendem.

- **Processo de auditoria:** A auditoria é o processo responsável em assegurar que todas as atividades e tarefas estão sendo realizadas corretamente. A norma cita que a organização deve conter um profissional qualificado capaz de propor constantes planos de melhorias, possibilitando a organização cumprir suas metas nos tempos certos com obtenção de sucesso.
- **Processo de resolução de problemas:** Este é o processo responsável por manter todo o ciclo de vida do projeto. Assim como na solução de problemas para o produto, a norma também apresenta um processo que solucione os problemas para os demais processos, diminuindo os riscos de prejuízos e possíveis desvios de planejamento. Uma atividade indispensável neste processo é a elaboração dos relatórios de contenção de erros que possibilitam a realização do estudo das causas e desencadeia uma análise de solução mais descritiva e de fácil acompanhamento.

Os processos de apoio são utilizados quando a organização deseja impor uma qualidade aparente no produto. Deve-se levar em conta que se torna indispensável realizar todos os procedimentos citados acima, visto que no mercado atual e competitivo, a qualidade se torna a propaganda do produto.

7.4.4 Processos organizacionais

A categoria de processos organizacionais contém os processos denominados operacionais para um melhor funcionamento da Fábrica de Software. Nesta categoria engajam-se atividades relacionadas principalmente ao gerenciamento e capacitação de pessoas, qualificando a estrutura da empresa para suportar a realização e administração de projetos de sistemas. Os processos organizacionais descritos pela norma ISO/IEC 12207, segundo a ABNT (1998) são:

- **Processo de gerência:** Neste processo é descrito a necessidade em implantar as atividades de gestão para processos e produtos nos projetos desempenhados pela Fábrica de Software. O processo é dividido em atividades primordiais como definição de escopo, planejamento, execução, controle, revisão, avaliação e por fim o fechamento. A norma descreve o processo de gerência como uma fonte para determinar se o projeto obterá fracasso ou sucesso em sua realização de acordo com sua administração.
- **Processo de infraestrutura:** Este processo tem como função designar uma estrutura compatível para adaptar um novo processo desenvolvido pela organização para o projeto abordado. A definição e compatibilização da infraestrutura permitem a integração de novas ferramentas, técnicas, padrões, ou aspectos mais casuais como hardware e software.
- **Processo de melhoria:** É o processo responsável para estabelecer, avaliar, medir, controlar e melhorar um processo componente do ciclo de vida de software. A

- **Processo de treinamento:** O treinamento é preconizado como a atividade de capacitar os profissionais para efetuar as atividades dos processos de maneira rápida e eficiente. A norma ISO/IEC 12207 estabelece o treinamento como uma técnica simples e indispensável, pois a qualidade de um software pode ser alcançada se os profissionais que o desenvolvem forem pessoas discriminadas e capacitadas. Alguns pontos que devem ser observados e inseridos durante o treinamento são o desenvolvimento de um material adequado e de instrutivo, além de um plano escalar de treinamento que possibilite a integração de todos os participantes lhes dando estímulos suficientes para que se conscientizem da melhoria que deve ser implantada a medida que o conhecimento necessário fora sendo obtido.

Os processos organizacionais são destituídos de maneira hierárquica de acordo com o fator matricial da Fábrica de Software. A frequência com que os processos são realizados corresponde ao quadro estrutural atual da empresa, que dependendo da situação, pode executá-los de forma rotineira ou somente em situações difíceis, quando é necessária a busca em atingir metas e resultados positivos.

7.4.5 Processo de Adaptação

A implantação das normas ISO nas organizações é culminante de várias análises e pesquisas em virtude da necessidade de melhorias nos processos e produtos para atingir um nível qualitativo superior a demanda de investimentos realizados em determinados intervalos de tempo. Muitas destas organizações que tentam implantar este contexto, não se adéquam, ou pelo menos, não utilizam de forma correta, os padrões instituídos pela ISO dificultando o aproveitamento necessário que é preciso ser adquirido para que surjam os resultados satisfatórios conforme os padrões estabelecidos nos documentos técnicos e guias de referências fornecidos durante a aquisição das referidas normas.

Observando a realidade das organizações em definirem ou melhorarem seus ciclos de desenvolvimento baseados nos preceitos da ISO/IEC 12207, a ISO integrou um processo auxiliar com o intuito de prover uma adequação conveniente a estrutura da organização a medida que as alterações forem sendo realizadas de forma que os processos existentes não sofram alterações relevante que provenham conseqüências indesejáveis [MACHADO 2006].

As atividades descritas para esse processo, segundo a ABNT (1998), são baseadas em práticas simples que sintetizam o conhecimento em melhor espécie da organização:

- **Identificação do ambiente do projeto:** Esta atividade provém vários fatores essenciais que precisam ser observados para que a terminologia da norma não se torne tão complexa para o ambiente em questão. Alguns pontos abordados são o modelo de ciclo de vida, os requisitos do sistema e do *software*, as políticas de qualidade atuais da organização, os procedimentos e estratégias planejadas, os tipos de sistema, produto ou serviço de software e a quantidade de pessoas ou partes envolvidas durante a implantação ou atualização da norma.
- **Solicitação de informações:** As informações que são decorrentes para o processo de implantação são primordiais durante a adaptação dos requisitos da norma para os ciclos

Comment [j196]: Culminante? Resultado fica melhor

Comment [j197]: plural

operacionais ou usuários facilita eventuais consultas, alterações ou melhorias provenientes durante a implantação ou atualização da norma.

- **Seleção de processos, atividades e tarefas:** Os processos, suas atividades e tarefas devem possuir documentos contendo todas as suas características e os respectivos responsáveis por desenvolvê-los e executá-los. Estes fatores contribuem para o controle risco, custos de implantação, estipulação de um cronograma viável, verificação de índice de desempenho, medição de processos, e a quantidade de pessoas ou partes envolvidas durante a implantação ou atualização da norma.
- **Documentação das decisões e motivos de adaptação:** Todo o plano de viabilidade adaptativa, as análises de custos, riscos, além das atividades mencionadas anteriormente, precisam ser catalogadas adjuntas as decisões da organização no proposto da aquisição e implantação da norma em seu ciclo de desenvolvimento existente.

O processo de adaptação busca conscientizar a organização para as inserções de melhorias de gestão para processos que precisam ser adequadas durante a implantação de uma norma ISO para processos de software. A padronização do ciclo de desenvolvimento não decorre somente de inserir, alterar ou remover processos e suas características, mas sim, de capacitar os profissionais responsáveis por mantê-los da necessidade em conhecer intensivamente todos os detalhes das atividades e tarefas para melhor gerir mudanças que não sucumbam em situações desastrosas ou com resultados insatisfatórios pela a falta de conhecimento necessário.

7.5 ISO/IEC 15504

As normas para Processos de Software apresentadas anteriormente ditam uma visão coesa da importância que tem em dispor de etapas bem definidas, qualquer projeto que possua uma complexidade de planejamento e quantidades relevantes de informações durante a definição de um sistema de gestão para qualidade e principalmente na configuração do ciclo de desenvolvimento dos processos traçados por uma organização que proponha metas e estratégias para o desenvolvimento de sistemas de informação.

A norma ISO/IEC 15504 complementa de certa forma, uma análise completa de todo o contexto qualitativo designado para a avaliação de processos de software para as organizações denominadas Fábricas de Software. Responsável por identificar e mapear individualmente e coletivamente os processos componentes de um projeto de sistemas, esta norma propõe aos engenheiros o desenvolvimento de modelos de acompanhamento de processos e subprocessos através de definições, requisitos e medições, adicionando níveis de capacitação e atributos, de forma que se torne possível obter o diferencial qualitativo submisso à medida que o sistema de informação proposto fora desenvolvido baseado em referências quantitativas postergadas da relação existente entre os processos do projeto e nível de detalhamento e conhecimento obtidos pela organização durante sua execução e manutenção [ISO 2004a].

Comment [j198]: prolix, tentar usar menos palavras para expresser idéias

Comment [j199]: num entedi nada, falta pontuação virgule e etc

7.5.1 Avaliação de processos

A avaliação e monitoramento de processos é algo indispensável para qualquer projeto. O impacto de melhorias causado por modelos como o CMMI, e outros modelos de melhorias para processos durante os últimos anos, despertou a busca de novos métodos e alternativas que melhorassem internamente a capacidade de maturação de uma organização. observando

Para isso, não bastou apenas utilizar modelos complexos como protótipos para novas práticas organizacionais, mas sim, harmonizar um *framework* genérico de iterações simples que representasse de forma mais direta e menos burocrática o entendimento dos processos de uma determinada organização, independente de porte, podendo tornar possível a realização de acompanhamento e aplicação de melhorias dos processos sem necessitar de constantes mudanças e retrabalhos [ISO 2004a].

A ideia de normatizar o *framework* partiu de um projeto idealizado na mesma época do surgimento do CMMI. Muitos engenheiros associavam os modelos tradicionais como “restritos e duvidosos” em relação a sua aplicação. Se aplicados corretamente em uma organização que possui processos bem definidos e caracterizados, resultariam em mudanças positivas bastante previsíveis, porém se adaptados a uma estrutura inconsistente, poderiam acarretar conseqüências indesejáveis que comprometeriam a estrutura da organização. Para a época, a indagação unânime: Como saber se é viável ou não aplicar o CMMI ou os outros modelos burocráticos para melhorias de processos (MPS)? Esta resposta está diretamente relacionada à administração de processos e subprocessos formadores da organização, definidos e mapeados a partir de técnicas e modelos previamente instituídos com o surgimento de um projeto denominado *SPICE*.

7.5.2 Projeto SPICE

O início da norma ISO/IEC 15504 remarca uma volta aos anos noventa, mais precisamente em 1991. O JTC1 seguia uma linha de pensamento que idealizava a normatização de conceitos que facilitassem a perspectiva de denominação e definição de características para processos. Em 1993, teve início o projeto intitulado *Software Process Improvement and Capability dEtermination (SPICE)*, que segundo Koscianski e Soares (2007) possuía três objetivos principais:

- Auxílio para referência e apoio ao projeto da norma;
- Execução de testes de campo;
- Obtenção de dados e experiências práticas para conscientização na importância para o surgimento de uma nova norma;

O projeto SPICE provinha a devida complexidade para formular posteriormente uma norma consistente devido aos constantes investimentos que estavam sendo realizados à medida que o projeto ganhava destaque com a síntese de bons resultados em várias organizações. Após vários *templates* expedidos pela ISO em parceria com a IEC, através de seus grupos de trabalhos, em 1998 foi publicada a versão inicial da norma. A versão destacava exclusivamente o software, baseando-se nos processos de ciclo de vida constituintes da ISO/IEC 12207, adicionando a integração do ciclo e suas fases para definição de referências e requisitos, principalmente com o intuito de facilitar a elaboração de documentações para as etapas componentes dos projetos de sistemas [KOSCIANSKI e SOARES 2007]

Em 2003, deu-se por encerrado o projeto alicerce da norma ISO/IEC 15504. O *SPICE Network* surgiu como fase final de transição para publicação da norma reunindo uma coleção de documentos que satisfizessem os requisitos determinados pela ISO para que no mesmo ano finalmente a norma ISO/IEC 15504 fosse realmente publicada. [KOSCIANSKI e SOARES 2007].

7.5.3 Estrutura da norma: Referência de processos

Após sua publicação em 2003, várias mudanças em relação à versão final do projeto *SPICE*

organização que necessitasse avaliar e melhorar os processos com o uso das práticas e preceitos estabelecidos pela mesma. Atualmente, a ISO/IEC 15504 está dividida em cinco partes descritas pela ISO (2004) conforme apresentadas na Tabela 7.3:

Tabela 7.3: Estrutura da norma ISO/IEC 15504
Fonte: Adaptado de [ISO 2004a]

Parte	Descrição
1	Conceitos e vocabulários da norma
2	Execução de avaliações para processos (<i>framework</i>)
3	Orientações para realizar as avaliações de processos
4	Orientação em uso para melhoria de processo e determinação de capacidade de processo
5	Contém um exemplo de aplicação

Observando a Tabela 7.3, nota-se que a contextualização da norma é subdividida em várias partes independentes. Como a norma estabelece critérios diversificados para avaliação de processos, a ISO instituiu inicialmente a ideia principal de definir uma estrutura (*framework*) para que os processos fossem mapeados e monitorados, e aos poucos foi estipulando práticas e melhorias que gerassem alternativas viáveis para administração e alterações de requisitos e necessidades que são inseridos no processo à medida que o projeto é executado.

As partes, então denominadas de 1 a 5, são caracterizadas com os seguintes preceitos: [ISO 2004b]

- Parte 1 (Conceitos e vocabulários): Apresenta uma introdução geral sobre os conceitos de processo e avaliação para processos, também destacando um glossário geral sobre como iniciar as avaliações e determinar os principais termos avaliativos relacionados para o referido processo.
- Parte 2 (Avaliação de Processos): Define os requisitos mínimos para a realização de uma avaliação que garanta coerência e boa granularidade para o processo. Nesta parte da norma, são efetuadas classificações de dados, destacando principalmente a implantação de responsabilidades, capacidades e o desenvolvimento de modelos que possibilitem o acompanhamento dos processos.
- Parte 3 (Recomendações de avaliação): Provêm recomendações para avaliação dos requisitos e aplicação de melhorias dos mesmos para o processo.
- Parte 4 (Recomendações para melhoria de processos e determinação de capacidades): O objetivo desta parte da norma é inserir continuamente a perspectiva de uma boa visibilidade para especificar precisamente as capacidades do processo, identificando os pontos fortes, pontos fracos e os riscos do processo selecionado em relação a um determinado requisito especial.
- Parte 5 (Contém um exemplo de aplicação): O intuito desta parte é apresentar um exemplar do *Process Assessment Model* (PRM) que sintetiza um modelo de referência de processos baseado no ciclo de vida de processos instituído pela norma ISO / IEC

Comment [j200]: reescrever isso, não precisa dizer os numeros

Assessment Model (PAM), onde um ou mais fatores demandem avaliação e medição de processos de acordo com planejamentos e cronogramas.

Para o desenvolvimento de sistemas de informação e áreas de software afins é utilizada apenas a Parte 2 da norma. Koscianski e Soares (2007) citam o objetivo desta parte da norma resumidamente com a seguinte frase:

“Nesta parte, diferentemente da versão antiga, a 15504 não mais define os processos, mas sim um conceito chamado modelo de referência de processo.”

Como destacado pelo autor, os modelos de análise e medição de processos evidenciam suas características de acordo com uma nova abordagem instituída pela ISO através da descrição das necessidades sobre os processos para possibilitar as avaliações dos requisitos instituídos pelas suas atividades e tarefas. Os chamados modelos de referência buscam planificar o macro processo, ou processos menores contidos, para idealizar um controle instável com o propósito de situar a organização e seus colaboradores de como executá-los e poder acompanhá-los em uma escala contínua para que as correções e melhorias possam ser identificadas e realizadas momentaneamente.

O *Process Reference Model* (PRM) é um modelo que deve ser desenvolvido pela organização com o intuito de referenciar os processos e tornar explícitas as principais necessidades almejadas durante sua execução em relação aos demais processos. Sua principal característica é descrever os requisitos dos processos priorizando o estabelecimento de alternativas que permitam sua avaliação em relação aos resultados esperados para cada processo e seus índices de correspondências durante uma fase ou etapa do processo à medida que os artefatos do projeto são produzidos decorrentes do planejamento efetuado anteriormente [ISO 2004b].

Através deste modelo de referência, uma organização pode dimensionar um projeto e estipular fielmente as iterações, os prazos, dentre outros fatores que influenciam no início, na continuação e na finalização do projeto, além de influenciar diretamente na dinamização das atividades e tarefas com ganhos de conhecimento perceptíveis sobre os processos, possibilitando em muitos casos, a diminuição da incidência de erros e propiciando a melhoria de funcionamento da estrutura organizacional para gerar fatores de qualidade que resultarão em boas práticas e resultados comprovadamente satisfatórios.

Para realizar as medições nos processos, o *Process Assessment Model* (PAM) foi designado para complementar as referências antes realizadas. O modelo de avaliação é semelhante ao PRM diferenciando apenas em relação a algumas características que são atribuídas para guiar um programa de monitoramento previsível a números e indicadores quantitativos de produção. A avaliação é desenvolvida baseada nas referências descritivas de dois tópicos essenciais que norteiam o acompanhamento dos processos [ISO 2004b]:

- **Práticas Base (*Base Practice*):** São as principais práticas realizadas durante o processo ou subprocessos formadores do projeto.
- **Artefatos Produzidos (*Working Products*):** São produtos resultantes da execução de determinada fase do processo ou subprocessos formadores do projeto.

O conjunto de todos esses fatores influencia diretamente na verificação das chamadas dimensões (processo e capacidade), permitindo contextualizar se os processos estão organizados e são executados, e o índice de evolução quantitativa para os mesmos. O PAM atribui uma escala de seis níveis de capacitação que identifica em que status de evolução atual

Comment [j201]: reescrever

Comment [j202]: ocultar

do CMMI, à medida que o processo é dimensionado e melhorado [KOSCIANSKI e SOARES 2007]. A ISO (2004b) classifica estes níveis em categorias conforme apresentados:

- 0 – Incompleto: O processo não é implementado, ou não alcança seu propósito planejado.
- 1 – Executado: O processo implementado alcança seu propósito, mesmo que de forma não criteriosa ou desorganizada.
- 2 – Gerenciado: O processo possui todo um controle de monitoração para que isso influa diretamente em seus produtos.
- 3 – Estabelecido: O processo é implementado sistematicamente e de forma garantida.
- 4 – Previsível: O processo é implementado verificando-se constantemente os resultados alcançados dentro dos limites de execução.
- 5 – Otimizado: O processo é adaptado para atingir seus resultados eficientemente.

Visto que as etapas de um processo podem ser ordenadas de forma diferente, a norma ISO/IEC 15504 dita que a maturidade só pode ser obtida para cada fase do processo ou subprocessos se os resultados para ele forem alcançados sem a necessidade de futuras revisões. Desde sua implementação inicial, determinada do nível 0, até atingir os objetivos de negócio, no nível 5, o processo é avaliado constantemente e logo documentado em seguida, propondo à organização apenas administrá-lo e torná-lo susceptível a possíveis mudanças que venham a surgir, como por exemplo, a alteração de cargos de pessoas e a inserção de novas tecnologias[ISO 2004b].

É importante salientar que a norma ISO/IEC 15504 não define os passos que devem ser seguidos para a elaboração dos modelos PRM e PAM. A norma cita apenas os componentes indispensáveis que devem ser desenvolvidos para um bom acompanhamento dos processos, dentre os quais se destacam a avaliação através de referências de necessidades (requisitos) e a estipulação de prazos para entrega de produtos (medição).

7.5.4 Dimensão de processos

A dimensão atribuída para os processos identifica a forma de como é estabelecida a organização do projeto, e sua execução em relação aos processos, caracterizando uma forma de como discernir um ou mais processos, contribuindo para que os engenheiros de software se situem dentro do projeto, o quanto de atividades já foi realizado e o que falta ser ainda finalizado.

A partir desta aplicação é possível estabelecer uma visão real do andamento dos processos e subprocessos durante as iterações definidas em um documento de escopo. A norma ISO/IEC 15504 classifica a dimensão de processo em cinco principais categorias, descritas por Koscianski e Soares (2007) da seguinte forma:

- Consumidor e fornecedor (CON): Define os processos que emanam a relação direta existente entre a fábrica de software e os consumidores, neste caso o cliente. Um dos processos condizentes a categoria é o levantamento de requisitos, buscando converter uma visão leiga do consumidor para uma visão técnica do fornecedor.

- Engenharia (ENG): Na categoria ENG, são inseridos os processos que designam o desenvolvimento do sistema. Alguns dos principais são o projeto de arquitetura, a codificação, testes e integração.
- Suporte (SUP): Os processos de suporte dão apoio aos demais processos do projeto. Nesta categoria, engajam-se reuniões, auditorias e revisões com intuito de angariar alternativas para solução de problemas.
- Administração (MAN): Nesta categoria, são adicionados os processos que abrangem aspectos genéricos do projeto. Um exemplo de um processo de administração indispensável para um projeto de software é documentação realizada ao longo de todo o projeto. Tanto nas fases iniciais, como coleta de requisitos e negociação entre fábrica e cliente, quanto ao término do projeto, com a liberação e implantação, são elaborados cronogramas e planos que tornam verídicas e registradas as atividades realizadas pelos engenheiros de software durante o projeto.
- Organização (ORG): Os processos organizacionais identificam os fatores que descrevem o funcionamento da empresa. Nesta categoria, são incluídos processos de gerência de conhecimento como infraestrutura e treinamentos.

O detalhamento de cada processo e suas classificações em categorias são atribuições previamente sugeridas pela norma. Cabe a organização preferir o meio mais viável para organizar e dimensionar os processos, estipulando como será a execução em relação aos requisitos descritos no PRM. A ISO/ICE 15504 não norteia os usuários sobre “como fazer”, mas sim, “o que fazer” durante o desenvolvimento dos modelos de referência e as metodologias que deve ser adotadas para a avaliação e medição dos processos.

7.5.5 Dimensão de capacidade

A organização e execução de processos de um projeto são dinamizadas através da dimensão de processos, tornando bastante implícita a perspectiva de execução dos cronogramas nos prazos certos. A dimensão de capacidade também avalia estes fatores, porém utilizando critérios mais específicos, semelhantes ao do CMMI [ISO 2004b].

Para dimensionar a capacidade intitulada para cada processo ou subprocesso, o PRM insere classificações para as atividades do processo, denominadas “atributos de processo”, permitindo avaliá-los em uma escala de cumprimento percentual. Cada atributo possui associado um indicador de descrição, no qual é atribuído um nível de contingência qualitativo, sendo apresentado como: Não atingindo, Parcialmente atingindo, Largamente Atingindo ou Totalmente atingido, conforme apresentado na Tabela 7.4:

Tabela 7.4: Atributos de capacidade da norma ISO/IEC 15504
Fonte: Adaptado de [ISO 2004b]

Escala	Nível	Porcentagem
N	Não atingido	0% a 15 % alcançado
P	Parcialmente atingido	15% a 50% alcançado
L	Largamente atingido	50% a 85% alcançado
F	Totalmente atingido	85% a 100% alcançado

Nota-se na tabela acima que há descrições para cada nível de evolução. No nível “N”, o

forma que foram planejadas, tornando sua execução sistemática para cumprir as metas estabelecidas. O “L” engloba em grande parte a manutenção e sustentabilidade do processo. As melhorias são atribuídas constantemente para influenciar no aumento de desempenho e correspondência das atividades, também podendo diminuir caso a administração não ocorra da maneira que deveria ser feita pelos profissionais. O “F” é o nível de refinamento do processo. Com todas as atividades concluídas dentro dos prazos definidos, não há falhas visíveis, destituindo confiança em sua aplicação e produzindo resultados visivelmente positivos.

7.5.6 Níveis de capacidade

Para facilitar o acompanhamento e visualização da transição entre os atributos que formam um processo, a norma ISO/IEC 15504 desenvolveu o mapeamento dos atributos em relação à escala de evolução dos níveis qualitativos definidos pela dimensão de capacidade.

O mapeamento consiste em alinhar em ordem crescente os atributos do processo justificando-os gradativamente em níveis de capacidade, ou seja, a cada novo atributo executado na ordem do projeto, o nível qualitativo posterior é atribuído sendo o nível qualitativo antecessor denominado para o próximo atributo. Koscianski e Soares (2007) demonstram, na Tabela 7.5, um exemplo da implementação dos níveis de capacidade:

Tabela 7.5: Atributos de capacidade da norma ISO/IEC 15504
Fonte: [Koscianski e Soares 2007, p. 163]

Atributos	Níveis de Capacidade				
	1	2	3	4	5
1.1	L ou F	F	F	F	F
2.1		L ou F	F	F	F
2.2		L ou F	F	F	F
3.1			L ou F	F	F
3.2			L ou F	F	F
4.1				L ou F	F
4.2				L ou F	F
5.1					L ou F
5.2					L ou F

processo utilizado como exemplo são destacados vários atributos representados por numerações. Cada atributo possui seus subitens e estes recebem capacidades através de níveis. A “escada” formada na tabela é constituída à medida que o projeto é realizado,

A ISO/IEC 15504 contribui em largos passos na obtenção da Qualidade para um Software. Comparando-se o planejamento de um projeto com definição de metas, requisitos, entre outros aspectos que englobam todo o conjunto de fatores sem o uso de pelo menos parte desta norma, há grande tendência em fracasso com consequências bastantes indesejadas.

7.6 Conclusões

O estudo da melhoria de processos com o uso de normas ISO é **comumente** relevante a complexidade existente no contexto de abordagem caracterizador das ditas normas. Os padrões fixos delimitam a adequação do uso destes artifícios, que compreendem **retornos garantidos** quando aplicados corretamente, apesar dos custos e investimentos que se fazem necessários no âmbito da obtenção de um reconhecimento qualitativo indispensável para os projetos de sistemas nas organizações.

Mesmo com tantos empecilhos, principalmente para as organizações de pequeno e médio porte, o uso das normas cresce a cada dia e os resultados obtidos despertam a atenção do mercado para a valorização dos princípios de responsabilidade, ética, compromisso, e principalmente respeito ao cliente, trazendo valores que antes não eram reconhecidos devido **a falta de uma administração consciente** que desmitificasse o conceito de auto-suficiência e trabalhasse realmente na expectativa de valorizar quem traz consigo um ponto definidor para o progresso ou fracasso dos produtos e serviços por elas geradas.

Este capítulo buscou de forma sucinta e objetiva apresentar os principais conceitos de algumas das principais normas para qualidade de processos de software utilizadas atualmente. Foram abordadas as entidades que regem essas normas, a ISO, o IEC e a ABNT, **breves histórias** de cada uma delas, as relações que as normas possuem com outros modelos de melhorias de processos, uma abordagem simples e direta de sistemas de gestão para qualidade, além do ciclo de vida para o desenvolvimento de software e alguns modelos de acompanhamento e avaliação de processos para obtenção de qualidade em processos de software.

Sugestões de leitura

Para se obter mais informações sobre as normas abordadas neste capítulo estão descritos alguns livros e documentos essenciais de autores renomados sobre os respectivos assuntos:

- Norma ISO 9001: 2008: Sistema de Gestão da Qualidade para operações de produção e serviços de Mello & Silva 2009
- Normas ISO 9001, ISO/IEC 12207 e ISO/IEC 15504: Qualidade de Software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2 ed. de Koscianski e Soares 2007
- Sistemas de Gestão para Qualidade de processos: Gestão da Qualidade 9ª edição

Comment [j203]: ocultar

Comment [j204]: ã?

Comment [j205]: Crase?

Comment [j206]: Breve histórico

- ISO/IEC 12207: NBR ABNT ISO/IEC 12207 ISO 12207 – Tecnologia da Informação – Processos de ciclo de vida de software. 1998
- ISO/IEC 15504: ISO/IEC 15504 Information Technology — Process Assessment — Part 2: Performing an Assessment. Versão 2.1

Tópicos de pesquisa

As pesquisas sobre os tópicos abordados neste capítulo são diversificadas pelas instituições mantenedoras dos padrões das normas. Visto que são conteúdos restritos a contratos e fechados para consulta pública, os avanços nas pesquisas se dão apenas de atualizações das correntes normas pela ISO provenientes de eventos, encontros, congressos, dentre outras oportunidades, onde são repassadas experiências de vários órgãos, sejam privados ou não, estipulando quais as principais dificuldades, diretrizes, restrições que precisam ser atualizadas.

No Brasil a pesquisa segue os mesmos rigorosos critérios intuídos pela ABNT. Vários associados formam o comitê da qualidade número vinte e cinco (<http://www.abntcb25.com.br/>) regendo as normas no formato NBR e auxiliando as empresas de consultorias no processo de certificação e implantação de normas ISO para gestão da qualidade.

Exercícios

1. Ao longo de todo o capítulo torna-se notável a importância que as normas exercem no contexto dos padrões que devem ser adotados pelas as empresas para que as mesmas se destaquem no mercado que demanda maior qualidade e praticidade e menor tempo e custo. Qual a importância de propor a adoção de normas ISO seja em caráter certificador ou não, nos dias atuais?
2. A implantação e manutenção de sistemas de gestão para qualidade em organizações com normas ISO, dentre elas a ISO 9001, envolve um macro planejamento desde a alta hierarquia aos colaboradores técnicos conforme apresentados na seção 7.3.1 em questão. Sabe-se que a definição de um sistema único e padronizado envolve todos os processos, suas atividades e tarefas, além da completa dedicação dos profissionais para com seu correto funcionamento. Esboce um pequeno índice de práticas que poderiam ser agregadas aos oito princípios da versão ISO 9000:2000 mencionadas na mesma seção que trariam melhorias significativas durante a implantação de um SGQ.
3. Baseado nas informações apresentadas neste capítulo, na seção 7.3.6, descreva resumidamente com suas palavras o processo de consultoria e implantação da ISO 9001:2008.
4. A ISO/IEC 90003 é um guia técnico complementar a ISO 9001 para Fábricas de Software. Explique cada atividade, de ciclo de vida e suporte, citando suas características e principais diretrizes para implantação.
5. O que você entende por “ciclo de vida de desenvolvimento” segundo a norma ISO/IEC 12207?
6. O processo de adaptação da norma ISO/IEC 12207 envolve algumas práticas administrativas essenciais que todas as organizações deveriam adotar no seu

Comment [j207]: Muito pobre, colocar os tópicos específicos de temas de pesquisa, junção de práticas e normas

Comment [j208]: Â?

Comment [j209]: Experiências de organizações, basta

esboço que associe essas práticas aos processos primários, organizacionais e de apoio.

7. O projeto *SPICE* surgiu com o intuito de amenizar as dificuldades de implantação provenientes do CMMI e outros modelos mais complexos da época que eram voltados para grandes organizações e exigiam altos conhecimentos sobre os conceitos de processos e sistemas de gestão. Faça uma pesquisa que descreva os propósitos principais almejados pelos engenheiros da época fazendo um comparativo deste projeto com as principais características que descrevem os níveis de maturação do CMMI.
8. Explique e diferencie os modelos de referência PRM e PAM.
9. O que você entende por dimensão de processo? Esta dimensão torna-se semelhante aos conceitos de ciclo de vida da ISO/IEC 12207? Aponte uma relação que poderia ser apresentada baseada nestas duas normas.
10. A dimensão de capacidade para processos, e os níveis de capacidade relacionados com esta dimensão, exercem total influência nas alterações de características dos atributos e atividades executadas pelos colaboradores. Quais fatores influenciam diretamente na passagem de nível gradativa que os processos alcançam a medida que são avaliados e melhorados?

REFERÊNCIAS

ABNT. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Histórico e descritivo das atividades.** Brasil 2009a. Disponível em: < http://www.abnt.org.br/press_kit.htm > Acesso em: 30 Mar. 2009

ABNT. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Comitê vinte e cinco.** Brasil 2009b. Disponível em: < <http://www.abntcb25.com.br> > Acesso em: 10 Set. 2009

ABNT. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO 9001 – Sistema de Gestão de Qualidade – Requisitos. 2ª Edição 2008a

ABNT. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO 12207 – Tecnologia da Informação – Processos de ciclo de vida de software. 1998

ABREU, Maurício et al. **Gerenciamento de Processos de Negócios: BPM Business Process Management.** 2 ed. São Paulo: Editora Érica, Inc, 2007.

CORTÊS, Mário L. Modelos de Qualidade de Software: Norma ISO 9000-3. Instituto de Computação da Universidade Estadual de Campinas UNICAMP. Campinas 2009. Disponível em: < <http://www.ic.unicamp.br/~cortes/inf326/> > Acesso em 14 Out. 2009

MARINHO, Euler Horta. Gestão da Qualidade de Software: ISO 9000-3. Departamento de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto. Ouro Preto 2009. Disponível em: < <http://eulerhm.googlepages.com/cea446-gestãodaqualidadedesoftware> > Acesso em 15 Out. 2009

Comment [j210]: Excluir links das referências

Comment [j211]: Padronizar referências "" nos títulos dos trabalhos

Acesso em: 15 Out. 2009

ISO. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *Joining In. Participating in International Standardization*. Suíça 2007. pp. 8. Disponível em: < http://www.iso.org/iso/joining_in_2007.pdf > Acesso em: 03 Set. 2009

ISO. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information Technology — Process Assessment — Part 1: Concepts and Vocabulary*. Versão 2.1 Suíça 2004a.

ISO. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information Technology — Process Assessment — Part 2: Performing an Assessment*. Versão 2.1 Suíça 2004b.

ISO. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *Key markers in ISO's history*. Suíça 2009a. Disponível em: < http://www.iso.org/iso/about/the_iso_story.htm > Acesso em: 04 Set. 2009

ISO. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, *ISO Standards*. Suíça 2009b. Disponível em: < http://www.iso.org/iso/iso_catalogue > Acesso em: 03 Set. 2009

FERREIRA, AURÉLIO BUARQUE DE HOLANDA FERREIRA. Novo Dicionário Eletrônico Aurélio versão 5.0 Positivo Informática. 2004
EMBRAPA, ISO 14000 Gestão Ambiental. Brasil 2009. Disponível em: < http://www.cnpma.embrapa.br/projetos/prod_int/iso_14000.html > Acesso em: 07 Set 2009.

IEC. INTERNATIONAL ELETROTECHNICAL COMISSION, *About the IEC*. Londres 2009a. Disponível em: < <http://www.iec.ch/helpline/sitetree/about> > Acesso em: 17 Set. 2009

IEC. INTERNATIONAL ELETROTECHNICAL COMISSION, *Types of IEC publicarion*. Londres 2009b. Disponível em: < <http://www.iec.ch/about/mission-e.html> > Acesso em: 17 Set. 2009

IEC. INTERNATIONAL ELETROTECHNICAL COMISSION *.ISO/IEC JTC 1 Information*. Genebra 2009c. Disponível em: http://isotc.iso.org/livelink/livelink/fetch/2000/2122/327993/755080/1054033/2541875/JTC001-N-9477_SD_2.pdf?nodeid=7846798&vernum=0 Acesso em: 17 Set 2009

FALBO, Ricardo de Almeida. *Qualidade de Processo Série ISO 9000: Tópicos Especiais em Qualidade de Software*. -Departamento de Informática da Universidade Federal do Espírito Santo. Vitória. 2007

MUSSI, Raimundo Nonato Fialho. FERREIRA, Meireluce da Silva. *Organismos Internacionais para a Ciência e Tecnologia: Coordenação de Assuntos Multilaterais - Secretaria Especial de Assuntos Internacionais - Ministério da Ciência e Tecnologia*. Brasília. 1988. pp.94.

KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de Software**: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2 ed. São Paulo: Novatec Editora, 2007

AMERICAN NATIONAL STANDARD INSTITUTE. *Introduction to ANSI*. Estados Unidos

JOINT TECHNICAL COMMISSION ONE. *Software and Engineering Subcommission 7*. 2008.
Disponível em: <<http://www.jtc1-sc7.org/>>

SPINOLA, Mauro de Mesquita. ISO 9000 para software. Textos acadêmicos Universidade Federal de Lavras. 2ª edição. 2005

CAMFIELD, Claudio Eduardo Ramos. GODOY, Leoni Pentiado. Análise do cenário das certificações da ISO 9000 no Brasil: um estudo de caso em empresas da construção civil em Santa Maria – RS. Universidade Federal de Santa Catarina. Florianópolis. 2003

FORTES, Marcel Menezes. Requisitos, documentos e registros mínimos para na NBR ISO 9001:2008. Arquivos de Qualidade da Petrobrás. 2ª revisão. 2009

MACHADO, Cristina F. Definindo Processos do Ciclo de Vida de Software Usando a Norma NBR ISO/IEC 12207 e Suas Ementas 1 e 2. Lavras: UFLA/FAEPE, 2006.

MATOS, Carlos Alberto Oliveira. O que você tem com ISO?. 2009. Disponível em: http://www.itapeva.unesp.br/docentes/carlos_alberto/qua4.pdf Acesso em: 17 Set. 2009

LIMA, Fabio Uchôa. Introdução ISO 9000: Versão 2000. São Paulo 2001. Acesso em: 17 Set 2009. Disponível em: http://novosolhos.com.br/site/arq_material/7431_8104.ppt

Eutech Instruments Water Analys Solutions. *How does ISO 9000:2000 compare to ISO 9000:1994?* Disponível em: http://www.eutechinst.com/iso9001vs_ver2000.pdf Acesso em 15 Set. 2009

Lloyd's Register Quality Assurance do Brasil. ISO 9000:2000. Disponível em: http://www.lrqg.com.br/certificacao/qualidade/iso9000_2000_02.asp Acesso em: 15 Set. 2009
Deutsche Gesellschaft zur Zertifizierung von Managementsystemen do Brasil. Revisão das normas ISO 9000 para 2000. Disponível em: <http://www.dqs.com.br/downloads/ApresentacaoISO90002000.pdf> Acesso em: 16 Set. 2009

SIMÕES, Arlete Rosemary. SILVA, César Augusto Villela. SILVA, Carlos Eduardo Sanches. TURRIONI, João Batista. O Impacto da Certificação ISO 9000 na Burocratização das Empresas. XXIII Encontro Nacional de Engenharia de Produção. 2003.

Target Engenharia e Consultoria. Facilitadores de informação. Publicada a nova versão ISO 9000:2005. Disponível em: http://www.target.com.br/portal/asp/Materia/Materia_dados_1.asp?materia=438 Acesso em: 20 Set. 2009

MARSHALL JUNIOR, Isnard. CIERCO, Agliberto Alves. ROCHA, Alexandre Varanda. MOTA, Edmarson Bacelar. LEUSIM, Sérgio. Gestão da Qualidade 9ª edição São Paulo 2008.

MELLO, Carlos Henrique Pereira. SILVA, Carlos Eduardo Sanches. TURRIONE, João Batista. SOUZA, Luiz Gonzaga Mariano. ISO 9001: 2008: Sistema de Gestão da Qualidade para operações de produção e serviços. 2009

Arquitetura e Agronomia do Espírito Santo – CREA-ES. 2007. Disponível em:
http://www.creaes.org.br/downloads/palestra_BSI_3.pdf Acesso em: 22 de Set. 2009

MARANHÃO, Mauriti. *ISO série 9000: manual de implementação*
versão 2000. 6. ed. Rio de Janeiro : Qualitymark, 2001.

MUTAFELIJA, Boris. STROMBERG, Harvey. *Systematic process improvement using ISO 9001:2000 and CMMI*. Canton Street Norwood, MA. 2003

Índice do Capítulo

8.1. INTRODUÇÃO 102

8.2. HISTÓRICO 103

8.3. CMMI 103

8.3.1.	REPRESENTAÇÕES DO MODELO CMMI	105
8.3.1.1.	REPRESENTAÇÃO POR ESTÁGIOS	105
8.3.1.2.	REPRESENTAÇÃO CONTÍNUA	107
8.3.1.3.	REPRESENTAÇÃO POR ESTÁGIOS X CONTÍNUA	109
8.3.2.	MÉTODO DE AVALIAÇÃO DO CMMI (SCAMPI)	109
8.3.2.1.	CONCEITO CENTRAL	110
8.3.2.2.	PARÂMETROS OBSERVADOS NO SCAMPI	110
8.3.2.3.	PRAZO E EXIGÊNCIA DE PESSOAL	110
8.3.2.4.	CARACTERÍSTICAS ESSENCIAIS DO MÉTODO DE SCAMPI	110
8.3.2.5.	MODOS DE USO	111
8.3.2.6.	DESCRIÇÃO DO MÉTODO	111

8.4. MPS.BR115

8.4.1.	REPRESENTAÇÃO DO MODELO MPS	116
8.4.1.1.	NÍVEL G – PARCIALMENTE GERENCIADO	116
8.4.1.2.	NÍVEL F – GERENCIADO	117
8.4.1.3.	NÍVEL E – PARCIALMENTE DEFINIDO	117
8.4.1.4.	NÍVEL D – LARGAMENTE DEFINIDO	118
8.4.1.5.	NÍVEL C – DEFINIDO	118
8.4.1.6.	NÍVEL B – GERENCIADO QUANTITATIVAMENTE	118
8.4.1.7.	NÍVEL A – EM OTIMIZAÇÃO	118
8.4.2.	MÉTODO DE AVALIAÇÃO DO MPS.BR (MA-MPS)	118
8.4.2.1.	PRAZO E EXIGÊNCIA DE PESSOAL	120
8.4.2.2.	DESCRIÇÃO DO MÉTODO	120

8.5. CMMI X MPS.BR 123

8.6. EXERCÍCIOS 124

8.7. SUGESTÕES DE LEITURA 125

8.8. TÓPICOS DE PESQUISA 125

8.9. REFERÊNCIAS 126

Capítulo

8

Modelos de Maturidade para Processos de Software: CMMI e MPS.BR

Audrey Vasconcelos, Lenildo Morais

Este capítulo tem como propósito definir, explicar e ilustrar um dos aspectos mais importantes no processo de construção de software: a maturidade dos processos de software em uma organização, visando a qualidade do produto gerado e a consequente satisfação dos seus clientes. Para isso, serão apresentados os principais modelos de referência, CMMI e MPS.BR. O primeiro trata-se de um modelo internacional, desenvolvido pelo SEI (*Software Enginner Institute*); e o segundo, um modelo nacional, criado de acordo com a realidade das empresas brasileiras, visando a melhoria do processo de software no Brasil. Tais modelos podem subsidiar as organizações que almejam aprimorar seus processos de desenvolvimento de software, tornando-se assim mais competitivas.

Introdução

A qualidade de software é um dos assuntos mais discutidos atualmente na comunidade de engenharia de software. A preocupação em produzir software com maior qualidade e mais eficiência surgiu na década de 70, porém, foi na segunda metade da década de 90 que novos conceitos e abordagens alcançaram maturidade e visibilidade.

Hoje em dia, com a intensa utilização de software, nas mais diversas áreas, vários projetos têm sido desenvolvidos para suprir essa demanda. Todavia, muitos deles não são produzidos seguindo um padrão; seus fabricantes, na maioria das vezes, enfatizam apenas em satisfazer as necessidades iniciais do cliente, suprimindo aspectos evolutivos do software, gerando preocupações consideráveis quanto à qualidade desses produtos.

Diante deste cenário, a área de desenvolvimento de software se tornou um nicho lucrativo para as empresas da área de Tecnologia da Informação. Buscando uma maior inserção no mercado de desenvolvimento de software, diversas corporações começaram a fazer grandes investimentos para desenvolver sistemas diferenciados com mais qualidade. Para isto, investiu-se também na melhoria do processo de desenvolvimento de software e passou-se a buscar a adoção de modelos de qualidade de software com reconhecimento internacional que pudessem certificar que os sistemas desenvolvidos pela organização são sinônimos de qualidade. Com isso, foram criados os modelos de qualidade de software que têm como objetivo garantir a qualidade do produto através da definição e normatização de processos organizacionais a serem

Desta forma, hoje, um dos maiores desafios das empresas produtoras de software é justamente mesclar os novos conceitos de engenharia de software com as práticas de qualidade preconizadas por esses modelos de processos de software.

Histórico

Há algum tempo a produção de software era feita sem nenhum processo definido. Não havia controle algum sobre possíveis manutenções ou possibilidade de integração com outros módulos. Logo, os sistemas antigos tinham um ciclo de vida curto e custos elevados. À medida que a complexidade dos projetos aumentava, novas tecnologias foram surgindo, e uma necessidade de mudança nos processos de produção foi percebida por empresas que identificavam custos altíssimos devido ao desenvolvimento de sistemas realizado com um comportamento completamente *ad-hoc*. Com isso, a engenharia de software foi em busca de um modelo que pudesse definir uma uniformidade para esses processos [KNEUPER 2009].

Com essa evolução, conceitos importantes surgiram gradativamente, tais como divisão de software, arquitetura *top-down* e *bottom-up*, diagramas e modelagens, conduzindo a engenharia de software ao estado atual. Com essa padronização, permitiu-se a oportunidade de mensurar alguns atributos de maneira mais precisa e segura, além de permitir mensurar o tamanho do software antes de sua construção. No entanto, mesmo com esse avanço, novos fatores surgiam aumentando a complexidade de produzir software.

A natureza dos negócios se diversificava bastante e novas tecnologias surgiam, modificando completamente definições obtidas anteriormente. Somados a estes fatores, sempre existiam projetos mal construídos, o que tornava a produção de software bem diferente da produção da engenharia comum. Enquanto os sistemas se tornavam maiores e a natureza dos negócios se multiplicava, a engenharia de software também se tornava mais abrangente [KNEUPER 2009].

Os modelos de qualidade foram criados para ser um guia destinado a melhorar os processos organizacionais e a habilidade destes em gerenciar o desenvolvimento, a aquisição e a manutenção dos produtos de software. Tais modelos apresentam uma visão própria, porém, unanimemente, todos destacam a importância de capacitação e desenvolvimento das habilidades do capital humano. O *Software Engineering Institute-SEI* criou um modelo de maturidade de software – o *Capability Maturity Model-CMM* que futuramente evoluiu para *Capability Maturity Model Integration-CMMI* – que verifica o nível de maturidade da empresa em relação ao seu processo, tendo como base algumas metodologias para desenvolvimento de software, como o RUP. Dessa forma, este modelo de maturidade tornou-se um domínio específico da computação para a avaliação de uma empresa de desenvolvimento de sistemas.

CMMI

Organizações de desenvolvimento de software vêm aumentando, ao longo do tempo, suas percepções relativas aos problemas que tipicamente são identificados durante a execução dos projetos, tais como: prazos e orçamentos não cumpridos, insatisfações dos clientes, produtos com erros, entre outros.

Há algum tempo, existe um consenso na comunidade de engenharia de software de que estes problemas estão, em grande parte, relacionados ao fato de que o desenvolvimento de software é muitas vezes realizado de forma “artesanal”; isto é, através de métodos improvisados pelos desenvolvedores, os quais, por sua vez, muitas vezes dependem mais de seu talento individual que de uma sólida formação que oriente suas atividades. [SOMMERVILLE 2003].

representações, divididas em níveis, priorizando de forma lógica as ações a serem realizadas. Quanto maior o nível, maior a maturidade da organização, o que pode se traduzir em maior qualidade do produto final, com maior previsibilidade em cronogramas e orçamentos.

O objetivo do CMMI é servir de guia para a melhoria de processos na organização, assim como para auxiliar a habilidade dos profissionais em gerenciar o desenvolvimento de aquisição e manutenção de produtos ou serviços de software, além de proporcionar a visibilidade apropriada do processo de desenvolvimento para todos os envolvidos no projeto.

Isto é particularmente importante em grandes projetos que possuem equipes envolvendo dezenas de pessoas, pois, sem o apoio desses modelos de maturidade de processos de software como o CMMI, torna-se ainda mais difícil manter o controle do projeto. Com a utilização de níveis, o CMMI descreve um caminho evolutivo recomendado para uma organização que deseja melhorar os processos utilizados para construção de seus produtos e serviços. Os níveis também podem resultar de classificações obtidas por meio de avaliações realizadas em organizações compreendendo a empresa toda (normalmente pequenas), ou grupos menores, tais como um grupo de projetos ou uma divisão de uma empresa [SEI 2006].

Quando uma organização atinge um nível de maturidade, considera-se que seus processos alcançaram uma determinada capacidade, ou seja, tem mecanismos que garantem a repetição sucessiva de bons resultados futuros relacionados principalmente à qualidade, custos e prazos. Com isso, compreende-se que o modelo em uma organização pode ser alcançado em etapas consecutivas, representando a idéia de maturidade (avaliada por estágios) da organização, ou de maneira contínua, onde é mensurada a capacidade em práticas individuais, conforme ilustrado na Figura 8.1.

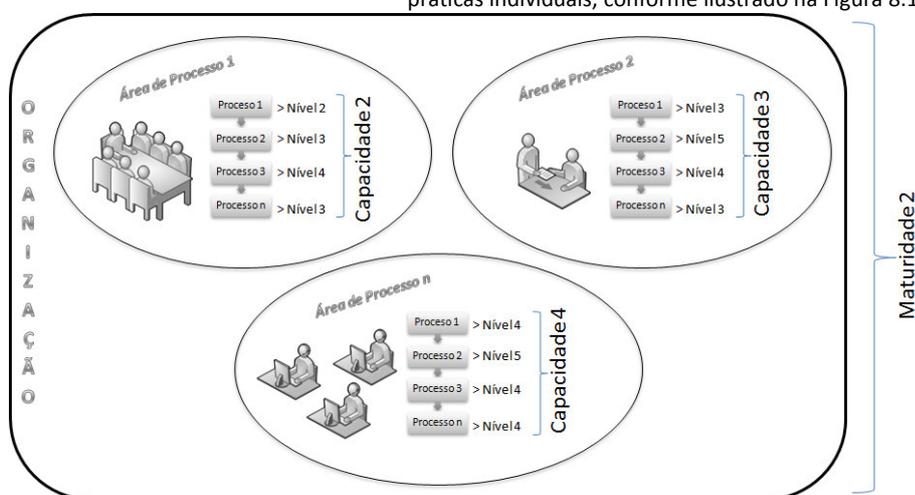


Figura 8.1. Capacidade e Maturidade Organizacional

O CMMI define cinco níveis de maturidade, onde no primeiro a empresa desenvolve sistemas baseando-se apenas na experiência dos recursos humanos da organização; e no último, há um processo organizado e flexível, com um planejamento eficiente e continuamente aprimorado. Para que uma empresa alcance níveis de maturidade superiores deverá cumprir metas compreendidas em cada área de processo (Process Area –

Existem quatro disciplinas presentes no modelo CMMI [KOSCIANKI 2007]:

- a) **Engenharia de Sistemas.** Utiliza uma abordagem interdisciplinar, com o objetivo de obter sucesso nos sistemas, envolvendo ou não software. Os engenheiros de sistemas propõem produtos e soluções por meio da análise, projeto, validação, teste, implementação, treinamento e suporte, de acordo com as necessidades e expectativas do cliente.
- b) **Engenharia de Software.** Foi criada com o objetivo de disciplinar a produção de software, em virtude dos problemas que haviam sido percebidos no desenvolvimento de software. Não se dedica somente aos processos técnicos de desenvolvimento de software, mas também às atividades de gerenciamento de projetos, desenvolvimento de ferramentas, métodos e teorias que dêem apoio à produção de software.
- c) **Desenvolvimento Integrado do Produto e do Processo.** Processos que utilizam a colaboração dos *stakeholders* para melhor satisfazer as expectativas e requisitos dos clientes. Esses processos são integrados aos outros existentes na organização.
- d) **Fontes de Aquisição (Gestão de Fornecedores).** Atua na aquisição de produtos sempre que o projeto necessitar de fornecedores que realizem funções específicas ou adicionem modificações em produtos específicos.

Representações do Modelo CMMI

A representação contínua permite que as organizações melhorem de forma incremental os processos correspondentes a uma ou mais *PAs*. A própria empresa seleciona em que áreas de processo ela será avaliada. A representação estagiada, por sua vez, permite que as organizações melhorem um conjunto de processos interrelacionados e, de forma incremental, tratem sucessivos conjuntos de *PAs*.

Representação por Estágios

Esta representação preocupa-se com os processos da organização como um todo. Oferecendo uma abordagem estruturada e sistemática para a melhoria de um estágio por vez. Attingir um estágio significa que uma estrutura de processo adequada foi estabelecida como base para o próximo estágio [SEI 2006].

As áreas de processo (*PAs*) são organizadas por níveis de maturidade – do nível “inicial” (nível 1) ao nível “em otimização” (nível 5) – que sugerem uma ordem para a implementação das áreas de processo. Cada nível possui várias *PAs*, e por sua vez, cada *PA* possui objetivos, práticas genéricas e específicas, assegurando assim uma base de melhoria adequada para o próximo nível de maturidade.

Na representação por estágios, quando uma organização atinge as práticas necessárias para estar em um nível, significa que atinge todos os requisitos necessários dos níveis imediatamente anteriores, conforme ilustrado na Figura 8.2 [KOSCIANKI 2007].

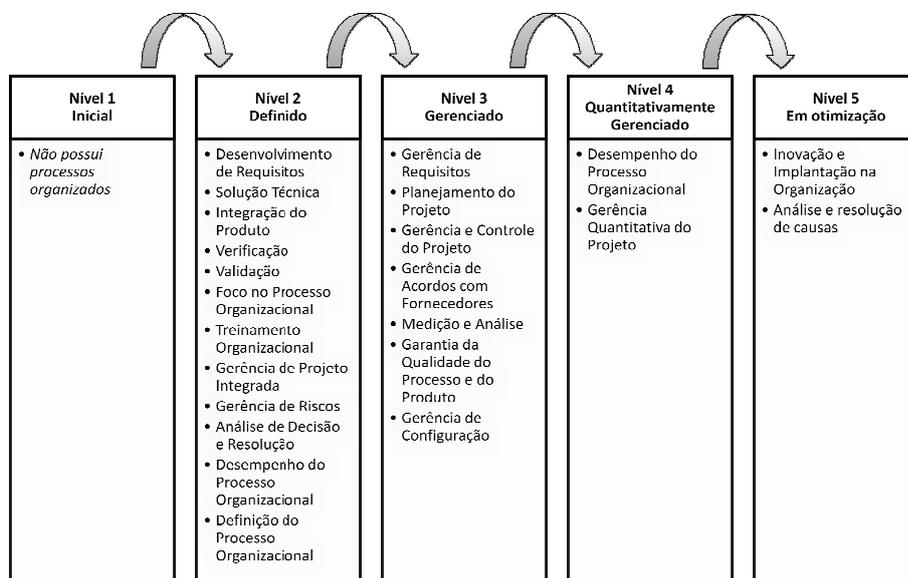


Figura 8.2. Áreas Chaves de Processo do CMMI – Representação por estágios

- Nível 1 – Inicial.** É o nível de maturidade CMMI mais baixo. Em geral, as organizações desse nível têm processos imprevisíveis que são pobremente controlados e reativos. Neste nível de maturidade não há *PAs*, os processos são normalmente *ad hoc* e caóticos, e a organização geralmente não fornece um ambiente estável.
- Nível 2 – Gerenciado.** Neste nível, os projetos da organização têm a garantia de que os requisitos são gerenciados, planejados, executados, medidos e controlados. Quando essas práticas são adequadas, os projetos são executados e controlados de acordo com o planejado. O gerenciamento de projetos é o foco principal deste nível.
- Nível 3 – Definido.** Nível em que todos os objetivos específicos e genéricos atribuídos para os níveis de maturidade 2 e 3 foram alcançados, os processos são mais bem caracterizados e entendidos e são descritos em padrões, procedimentos, ferramentas e métodos. O foco neste nível é a padronização do processo.
- Nível 4 – Quantitativamente Gerenciado.** Neste quarto nível, os objetivos específicos e genéricos atribuídos para os níveis de maturidade 2, 3 e 4 foram alcançados e os processos são medidos e controlados. O foco neste nível é o gerenciamento quantitativo.
- Nível 5 – Em Otimização.** É o nível mais alto de maturidade CMMI, onde uma organização atingiu todos os objetivos específicos atribuídos para os níveis de maturidade 2, 3, 4 e 5. Os processos são continuamente aperfeiçoados, baseados em um entendimento quantitativo em que a variação de um processo existe devido às interações normais e presumidas entre os componentes desse processo. Este nível de

Representação Contínua

Na representação contínua, o enfoque ou componentes principais são as áreas de processo. Existem metas e práticas de dois tipos: específicas a uma determinada área de processo e genéricas, aplicáveis indistintamente a todas as áreas de processo. Nesta representação, as áreas de processo são agrupadas por categorias afins. Os perfis de capacitação representam caminhos de melhoria indicando a evolução para cada uma das áreas. Em cada área de processo, os objetivos e as práticas específicas são listados, seguidos por objetivos e práticas genéricas, conforme ilustração da Figura 8.3.

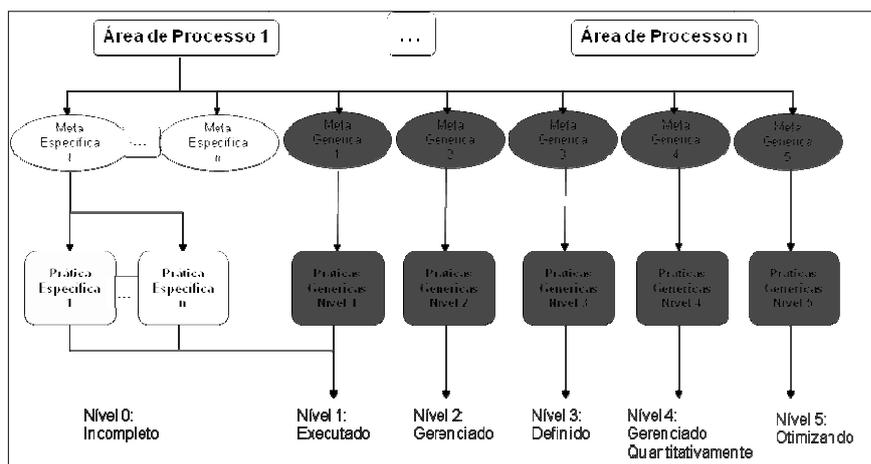


Figura 8.3. Tipos de Metas e Práticas da Representação Contínua

Muitos aspectos da representação contínua são agrupados em quatro categorias: Gerência de Processos, Gerência de Projeto, Engenharia e Suporte. As áreas de processo relativas à categoria de Gerência de Processos contêm atividades relacionadas para definir, planejar, implantar, monitorar, controlar, medir e melhorar processos. As áreas de processo relativas à categoria de Gerência de Projeto contêm as atividades de planejar, monitorar e controlar o projeto. A categoria de Engenharia refere-se às atividades de desenvolvimento de sistemas de software. As atribuições de fornecer suporte ao desenvolvimento e à manutenção de produtos são relativas à categoria de Suporte, conforme ilustrado na Figura 8.4.

Gerência de Processos	Gerência de Projetos	Engenharia	Suporte
<ul style="list-style-type: none"> • Foco no Processo • Definição de Processos • Treinamento • Desempenho de Processos • Inovação e Implantação 	<ul style="list-style-type: none"> • Planejamento de Projeto • Controle e Monitoração de Projeto • Gerência de Acordos com Fornecedores • Gerência de Projeto Integrada • Gerência de Riscos • Integração de Equipe • Integração de Fornecedores • Gerência Quantitativa de Projeto 	<ul style="list-style-type: none"> • Gerência de Requisitos • Gerência de Desenvolvimento • Solução Técnica • Integração de Produto • Verificação • Validação 	<ul style="list-style-type: none"> • Gerência de Configuração • Garantia de Qualidade de Produto e Processo • Medida e Análise • Análise de Decisão e Resolução • Ambiente Organizacional para Integração • Resolução e Análise de causas

Figura 8.4. Áreas de Processo do CMMI – Representação Contínua

A partir da avaliação e do atendimento dessas práticas e metas é possível classificar o nível de capacidade de cada área de processo, em uma escala de 0 a 5 [KNEUPER 2009]:

- Nível 0 – Incompleto.** Um processo é parcialmente realizado ou não, onde um ou mais objetivos específicos do processo não são satisfeitos.
- Nível 1 – Realizado.** Um processo realizado satisfaz todos os objetivos específicos da área de processo e produz algum trabalho.
- Nível 2 – Gerenciado.** Um processo de capacidade nível 2 é um processo realizado (nível 1) que também é planejado e executado de acordo com políticas pré-definidas. Emprega pessoas hábeis com os recursos adequados para produzir saídas adequadas, envolve os *stakeholders* principais e é monitorado, controlado, revisto e avaliado quanto à aderência à sua descrição. A gerência do processo é relacionada com a realização de objetivos específicos estabelecidos para o processo, como custo, cronograma e qualidade.
- Nível 3 – Definido:** Um processo definido é um processo gerenciado e ajustado para o conjunto padrão de processos da organização de acordo com suas políticas de conduta. Esse conjunto é estabelecido e melhorado com o tempo e descreve os elementos fundamentais de processos que são esperados nos processos definidos.
- Nível 4 - Gerenciado quantitativamente:** Um processo neste nível é definido e controlado com a ajuda de técnicas quantitativas e estatísticas. A qualidade e o desempenho do processo são compreendidos em termos estatísticos e são geridos durante sua vida. Objetivos quantitativos para qualidade e desempenho de processos são estabelecidos e usados como critério na gerência do processo.
- Nível 5 – Em otimização:** Um processo em otimização é gerenciado quantitativamente, alterado e adaptado para atender aos objetivos de negócio atuais e projetados. Tal processo enfatiza a melhoria contínua do desempenho do processo através de aprimoramentos tecnológicos inovadores e incrementais, selecionados com base em uma compreensão quantitativa de sua contribuição esperada à obtenção da melhoria

Representação por Estágios x Contínua

A representação contínua usa níveis de capacitação para medir a melhoria de processos, enquanto a representação por estágios utiliza níveis de maturidade para medir a melhoria de capacidade da organização. A principal diferença é a forma como cada representação é aplicada. Os níveis de capacidade são aplicados na melhoria de processos de cada área de uma organização. Existem seis níveis de capacitação, numerados de 0 a 5, onde cada nível possui um com objetivo geral e um conjunto de práticas gerais e específicas.

Na representação por estágios, os níveis de maturidade não servem para analisar áreas do processo, mas sim para indicar melhorias na organização como um todo. Ao fazer a avaliação de uma organização, é possível mapear os valores de capacidade do processo para maturidade organizacional. Uma organização que apresente nível 2 para todas as áreas de processo correspondentes ao nível de maturidade 2 é classificada nesse nível. Para todos os níveis superiores de maturidade, de 3 a 5, exige-se um nível de capacidade mínimo igual a 3 para as áreas de processo correspondentes ao nível pretendido.

Se não se sabe por onde começar e quais processos escolher para serem melhorados, a representação por estágios é uma boa opção. Ela fornece um conjunto específico de processos para melhorar em cada estágio, determinado por mais de uma década de experiência e pesquisas em melhoria de processo. Todavia, existem três categorias de fatores que podem influenciar na decisão de qual representação adotar [SEI 2006]:

- a) **Estratégico.** Se uma organização com foco em linha de produto decidir melhorar seus processos na organização como um todo, pode ser mais bem atendida pela representação por estágios, uma vez que a representação por estágios auxilia na escolha dos conjuntos de processos onde focar a melhoria. Por outro lado, a mesma organização pode optar por melhorar processos por linha de produto. Neste caso, ela pode escolher a representação contínua – e uma classificação diferente de capacidade pode ser obtida na avaliação de cada linha de produto. A consideração mais importante a ser feita é a identificação dos objetivos estratégicos a serem apoiados pelo programa de melhoria de processo e a forma como esses objetivos estratégicos se alinham às duas representações.
- b) **Culturais.** Estão relacionados com a capacidade da organização em implantar um programa de melhoria de processo. Por exemplo, uma organização pode escolher a representação contínua se sua cultura corporativa basear-se em processos e for experiente em melhoria de processo. Já uma organização pouco experiente em melhoria de processo pode escolher a representação por estágios, uma vez que essa representação fornece orientações adicionais sobre a seqüência em que as mudanças devem ocorrer.
- c) **Legado.** Caso uma organização tenha experiência com outro modelo que utiliza uma representação por estágios, pode ser mais prudente continuar utilizando essa representação no CMMI, principalmente se já investiu e implantou processos associados à representação por estágios. O mesmo raciocínio pode ser aplicado para a representação contínua.

Método de Avaliação do CMMI (SCAMPI)

O SCAMPI é um método utilizado para avaliação do modelo de referência CMMI. Ele foi projetado para atender todas as exigências de uma avaliação, podendo ser também utilizado para o modelo ISO/IEC 15504. O SCAMPI está atento aos seguintes aspectos:

- Grau de relacionamento que estes pontos positivos e negativos tem com o modelo CMMI;
- Priorização de planos de melhoria;
- Concentração nas melhorias;

A consolidação dos resultados das avaliações executadas é mantida pelo *SEI*. Tais dados são registrados de forma a catalogar os perfis de maturidade das organizações já avaliadas. Este perfil é baseado em avaliações providas por profissionais treinados do *SEI*, sendo atualizado semestralmente [SCAMPI 2001].

8.3.2.1. Conceito Central

O SCAMPI trabalha com uma agregação de evidência, ou seja, são considerados instrumentos como apresentações, documentos e entrevistas organizacionais, onde a equipe de avaliação observa, ouve e lê aspectos que darão respaldo na atribuição das notas e em resultados preliminares. Tais resultados são validados pela unidade organizacional antes de se tornarem resultados finais.

O planejamento é uma atividade extremamente crítica para execução do SCAMPI. Todas as fases e atividades do processo surgem de um plano bem elaborado e desenvolvido pelo condutor da avaliação, em relação aos membros da organização avaliada [SCAMPI 2001].

8.3.2.2. Parâmetros observados no SCAMPI

O SCAMPI procura ajustar seu método a finalidade e necessidades da organização, ou seja, o tamanho da empresa, a quantidade e o porte dos projetos, o tamanho da equipe de avaliação, a quantidade de entrevistas realizadas, tempo de preparação e custos, são parâmetros que devem ser documentados no plano de avaliação com o objetivo de respaldar sua execução.

A garantia de que as exigências do método serão atendidas é dada pelo líder da avaliação SCAMPI [SCAMPI 2001].

8.3.2.3. Prazo e Exigência de Pessoal

Três meses é o tempo considerado adequado para a realização de uma avaliação SCAMPI. Neste prazo estão incluídos planejamento, preparação e execução [Itaborahy et. al 2005].

Os atores que participam da execução de uma avaliação SCAMPI, são: o patrocinador, o líder da equipe de avaliação, o coordenador da unidade organizacional (OUC), os participantes selecionados e os membros da equipe de avaliação. Normalmente os participantes podem gastar três horas cada para realizar sessões de validação. Entretanto, o OUC pode gastar até três semanas de esforço de tempo para ajudar a equipe e a organização a preparar-se para administrar a avaliação.

8.3.2.4. Características essenciais do Método de SCAMPI

- Precisão.** Avaliações são reflexos da maturidade/capacidade da organização e podem ser comparadas pelas organizações.
- Repetibilidade.** Uma avaliação pode ser consistente com outra avaliação em condições análogas. Isto é outra avaliação com parâmetros idênticos produzirá resultados consistentes e semelhantes.
- Custo/Recurso Efetivos.** Eficiência em termos de custo de pessoas. O investimento da organização é considerado na obtenção do resultado da avaliação, incluindo os recursos da organização, o impacto nos projetos avaliados e a equipe de avaliação.

- d) **Gerenciamento dos Resultados.** Os resultados da avaliação têm impacto direto no contexto de melhoria de processo interno, escolha de fornecedores e monitoração do processo, além de ser utilizados pelo patrocinador como apoio na tomada de decisões.

8.3.2.5. Modos de Uso

Uma avaliação é um exame de um ou mais processos realizado por uma equipe de profissionais treinados que usam um modelo de referência com o objetivo de determinar os pontos positivos e negativos do processo. Normalmente ela é aplicada no contexto de melhoria de processos ou avaliação de capacidade. O termo “avaliação” é usado para descrever aplicações nestes contextos, tradicionalmente, conhecidas como *assessments* e *evaluations*.

Basicamente a diferença entre os termos *assessments* e *evaluation* é que *assessments* é uma avaliação que uma organização faz de si mesma com o objetivo de melhorar seu processo.

O termo *Assessments* origina-se da motivação interna das organizações em iniciarem ou continuarem com os programas de melhoria de processo. *Evaluations* é uma avaliação na qual um grupo externo entra na organização e examina seus processos respaldando-a para uma decisão relativa ao futuro do negócio. Elas são motivações tipicamente externas para as organizações cristalizarem o processo de melhoria [Almeida 2007].

Normalmente as avaliações SCAMPI são executadas de três formas, conforme descrito abaixo:

- a) **Melhoria de Processo Interno.** Organizações utilizam métodos de avaliação para checar seus processos internos, medindo o progresso de um programa implementado. Desta forma, a avaliação do SCAMPI complementa outras ferramentas para implementar as atividades de melhoria de processo.
- b) **Seleção do Fornecedor.** Os resultados da avaliação são, muitas vezes, utilizados para escolher os fornecedores. Os resultados são usados considerando o risco de um contrato com um fornecedor.
- c) **Monitoração do Processo.** O método de avaliação também é usado no monitoramento de processos, ou seja, os resultados da avaliação são usados para ajudar a organização no seu contrato, permitindo priorizar esforços baseado na observação dos pontos positivos e negativos dos processos da organização.

8.3.2.6. Descrição do Método

O SCAMPI consiste em três fases e onze processos essenciais, onde cada fase está descrita nas tabelas abaixo [Almeida 2007]:

Tabela 8.1. Fase 1 - Plano e Preparação para a Avaliação [Almeida 2007]

Processo	Propósito	Atividades
1.1 Análise das Exigências	Entender as necessidades empresariais da unidade organizacional para qual a avaliação está sendo solicitado. O líder da equipe da avaliação coleccionará informação e ajudará o patrocinador da avaliação a emparelhar os objetivos da avaliação com os objetivos empresariais da organização.	1.1.1 Determinar os Objetivos da Avaliação 1.1.2 Determinar as Restrições da Avaliação 1.1.3 Determinar a Extensão da Avaliação 1.1.4 Determinam as Saídas 1.1.5 Obter Comprometimento para Entrada da Avaliação
1.2 Desenvolvimento do Plano de Avaliação	Documentar as exigências, acordos, estimativas, riscos, método, seguimento, e considerações da prática (por exemplo, horários, logísticas e informação contextual sobre a organização) associado com a avaliação. Obter, registrar e fazer visível a aprovação do patrocinador do plano de avaliação.	1.2.1 Método de Seguimento 1.2.2 Identificar os Recursos Necessários 1.2.3 Determinar Custo e Horário 1.2.4 Planejar e Administrar a Logística 1.2.5 Documentar e Administrar os Riscos 1.2.6 Obter Comprometimento do Plano de Avaliação
1.3 Seleção e Preparação da Equipe	Assegurar que uma equipe experiente está disponível, apropriadamente qualificado e preparado executar o processo de avaliação.	1.3.1 Identificar o Líder da Equipe 1.3.2 Selecionar os Membros da Equipe 1.3.3 Preparar a Equipe
1.4 Obtenção e Análise Inicial das Evidências Objetivas	Obter informação que facilite a preparação específica da avaliação. Obter dados em práticas de modelo usado. Identificar áreas de assunto potenciais, intervalos ou riscos para ajudar no refinamento do plano. Adquirir entendimento preliminar das operações e processos da unidade organizacional.	1.4.1 Preparam os Participantes 1.4.2 Administrar os Instrumentos 1.4.3 Obter Evidência Objetiva Inicial 1.4.4 Inventário da Evidência Objetiva
1.5 Preparação para Coleção da Evidência Objetiva	Planejar e documentar a estratégia de coleção específica de dados incluindo as fontes de dados, ferramentas, tecnologia a ser usada e contingências para administrar o risco de dados insuficientes.	1.5.1 Executar Revisão de Disponibilidade 1.5.2 Preparar Plano de Coleta de dados 1.5.3 Replanejar a Coleção de dados (se necessário)

Tabela 8.2. Fase 2 - Administrando a Avaliação [Almeida 2007]

Processo	Propósito	Atividades
2.1 Examinando a Evidência Objetiva	Coletar informações sobre as práticas implementadas dentro da unidade organizacional e relacionar os dados resultantes com referência no modelo. Executar a atividade conforme o plano de coleta de dados. Efetuar ações corretivas e revisar o plano de coleta de dados se necessário.	2.1.1 Examinar Evidência Objetiva vindas dos Instrumentos 2.1.2 Examinar Evidência Objetiva vindas das Apresentações 2.1.3 Examinar Evidência Objetiva vinda dos Documentos 2.1.4 Examinar Evidência Objetiva vindas das Entrevistas
2.2 Verificação e Validação da Evidência Objetiva	Verificar as práticas de implementação da unidade organizacional para cada momento. Validar os resultados preliminares, descrevendo intervalos na implementação das práticas do modelo. Cada implementação de cada prática é verificada de modo que pode ser comparado com as práticas do CMMI e a equipe caracteriza até que ponto as práticas do modelo está implementado. Intervalos de implementação de prática é capturado e validado com os membros da unidade organizacional. Exemplo de implementações de práticas do modelo podem ser destacadas como forças para ser incluído nas saídas da avaliação	2.2.1 Verificar a Evidência Objetiva 2.2.2 Caracterizar a Implementação das Práticas do Modelo 2.2.3 Validar os Intervalos de Implementação da Prática
2.3 Documentação da Evidência Objetiva	Criar registros duradouros da informação reunindo por identificação e Então consolidando em notas, transformando os dados em registros que documentam a implementação da prática, como também as forças e as fraquezas.	2.3.1 Compreender/Revisar/Identificar as Notas 2.3.2 Registrar a Evidência Objetiva Presente/Ausente 2.3.3 Documentar a Prática de Implementação de Intervalos 2.3.4 Revisar e Atualizar o Plano de Coleta de dados
2.4 Geração dos Resultados da Avaliação	Satisfazer a meta do índice baseada, em grande parte, na implementação das práticas ao longo da unidade organizacional. Até que ponto a implementação da prática é determinada/julgada baseado em dados validados (por exemplo, os três tipos de evidência objetiva) colecionado por amostra representativa da unidade organizacional. O índice de nível de capacidade e/ou nível de maturidade é conduzida algoritmicamente pela satisfação da meta do índice.	2.4.1 Mostrar os Resultados e Metas do Índice 2.4.2a Determinar o Nível de Capacidade da Área de Processo 2.4.2b Determinar a Satisfação da Área de Processo 2.4.3a Determinar o Perfil da Capacidade 2.4.3b Determinar o Nível de Maturidade 2.4.4 Documentar os Resultados da Avaliação

Tabela 8.3. Fase 3 - Relatório do Resultado [Almeida 2007]

Processo	Propósito	Atividades
3.1 Entrega dos Resultados da Avaliação	Prover os resultados da avaliação acreditáveis que podem ser usados para o encaminhamento de ações. Representa as forças e as fraquezas dos processos em uso na ocasião. Provê o índice (se planejado para) que com precisão reflete o nível/maturidade da capacidade dos processos em uso.	3.1.1 Apresentar os Resultados Finais 3.1.2 Administrar as Sessões Executivas 3.1.3 Plano para os Próximos Passos
3.2 Arquivar e Empacotar a Avaliação	Preservar os dados importantes e registros da avaliação, e dispor de materiais sensíveis de maneira apropriada.	3.2.1 Colecionar a Lições Aprendidas 3.2.2 Gerar o Registro da Avaliação 3.2.3 Prover o Retorno da Avaliação ao Assitente do CMMI 3.2.4 Arquivar e/ou Dispor de Artíficos Fundamentais para tal

MPS.BR

O MPS.BR é um programa que foi criado em 2003, de acordo com a realidade de empresas brasileiras, com o objetivo de propor um modelo de processo para alcançar a Melhoria do Processo de Software Brasileiro [KOSCIANKI 2007].

O Modelo MPS.BR estabelece não somente um modelo de processos de software, mas também um método de avaliação e um modelo de negócio para subsidiar as empresas brasileiras que desenvolvem software. Este modelo foi elaborado com base nas normas internacionais ISO/IEC 12207 (que atualmente encontra-se na versão 2008) e ISO/IEC 15504-2 (veja o capítulo XXX) e no *Capability Maturity Model Integration-CMMI* (veja a seção 0) [SOFTEX 2009].

A Figura 8.5 ilustra a estrutura do Modelo MPS.BR que possui três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS).

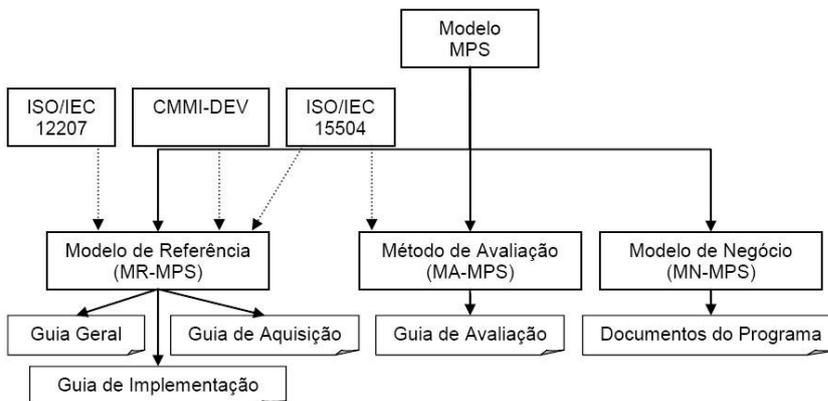


Figura 8.5. Estrutura do Modelo MPS.BR [SOFTEX 2009]

O Modelo de Referência (MR-MPS) possui três guias:

- Guia Geral.** Define todos os níveis de maturidade, processos e atributos de processo. Os níveis de maturidade definem o grau de evolução da implantação dos processos na organização em uma escala de sete níveis que inicia no nível G e vai até o nível A. Estes níveis serão detalhados na seção 8.4.1.
- Guia de Aquisição.** Descreve boas práticas para aquisição de software e serviços correlatos (S&SC) com foco na satisfação da necessidade do cliente. Para isso, segue um processo específico com as seguintes atividades: “Preparação para aquisição”, “Seleção do fornecedor”, “Monitoração do contrato” e “Aceitação pelo cliente”.
- Guia de Implementação.** Sugere formas de implementar cada um dos níveis descritos no Guia Geral e como uma unidade organizacional que realiza aquisições de software também pode implementar o Modelo MPS.

O Método de Avaliação (MA-MPS) contém um processo de avaliação que descreve um conjunto de atividades e tarefas para verificar a maturidade da unidade organizacional na execução dos seus processos de software. Este Método de Avaliação será detalhado na seção

O Modelo de Negócio (MN-MPS) descreve regras de negócio para: implementação do Modelo de Referência (MR-MPS); avaliação seguindo o Método de Avaliação (MA-MPS); organização de grupos de empresas para implementação e avaliação do MPS.BR; certificação de Consultores de Aquisição de S&SC, segundo o Guia de Aquisição do MPS.BR; realização de cursos, provas e workshops do MPS.BR, e para treinamento de pessoas no Modelo MPS [SOFTEX 2009].

8.4.1. Representação do Modelo MPS

Conforme mencionado anteriormente, assim como no CMMI, o modelo MPS define uma escala de níveis de maturidade, que vão do nível G ao nível A. Com isso, para que uma organização esteja em conformidade com um determinado nível, deverá realizar um grupo de processos específicos do nível corrente (que indica a maturidade que a organização pretende alcançar), bem como os grupos de processos do nível imediatamente anterior. Como por exemplo, para uma organização ser considerada em conformidade com o nível C, deverá ter implementado todos os processos dos níveis G, F, E, D e C.

Os processos são descritos em termos de propósito (que descrevem o objetivo geral a ser atingido durante a sua execução) e resultados esperados (que estabelecem os resultados a serem obtidos com a sua efetiva implementação).

Por sua vez, cada processo possui um conjunto de atributos que correspondem à sua capacidade, indicando o grau de refinamento e institucionalização com que este processo é executado na organização. Além disso, através destes atributos é possível mensurar o estado do processo, como por exemplo, medir se um processo é executado (atinge seu propósito), se a execução do processo é gerenciada, e assim por diante, inclusive identificar se as mudanças na definição, gerência e desempenho do processo têm impacto efetivo para o alcance dos objetivos relevantes de melhoria do processo, ou seja, se o processo é otimizado continuamente.

A representação de cada nível de maturidade com seus respectivos processos pode ser visualizada na Figura 8.6, excetuando-se os níveis A e B, pois estes não possuem processos específicos. A seguir, cada processo será detalhado segundo o Guia Geral do MPS.BR [SOFTEX 2009] [KOSCIANKI 2007].

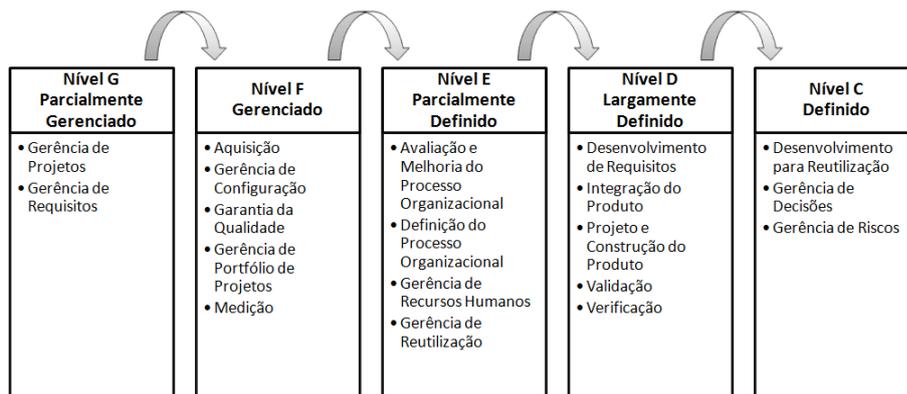


Figura 8.6. Processos do Modelo MPS.BR

O nível G do Modelo MPS é composto pelos processos “Gerência de Projetos” e “Gerência de Requisitos”, com os seguintes propósitos:

- a) **Gerência de Projetos.** Planejar as atividades, recursos e responsabilidades do projeto, bem como controlar o andamento do projeto. O propósito deste processo evolui à medida que a organização evolui na escala de maturidade (sobe de nível). Assim, a partir do nível E, alguns resultados evoluem e outros são incorporados, permitindo que a gerência de projetos possua um processo definido para o projeto e planos integrados. No nível B, a gerência de projetos absorve um enfoque quantitativo, transparecendo a alta maturidade que se espera da organização. E com isso, mais uma vez, alguns resultados evoluem e outros são incorporados.
- b) **Gerência de Requisitos.** Gerenciar os requisitos do produto do projeto, inclusive dos seus componentes, e identificar inconsistências que possam existir entre: requisitos, planos do projeto e produtos de trabalho.

8.4.1.2. Nível F – Gerenciado

O nível F do Modelo MPS, além dos processos do nível anterior, é composto pelos processos “Aquisição”, “Gerência de Configuração”, “Garantia da Qualidade”, “Gerência de Portfólio de Projetos” e “Medição”, com os seguintes propósitos:

- a) **Aquisição.** Gerenciar a aquisição de produtos que atendam às necessidades determinadas pelo adquirente.
- b) **Gerência de Configuração.** Estabelecer e manter a integridade de todos os artefatos produzidos e possibilitar que sejam acessados por todos os envolvidos.
- c) **Garantia da Qualidade.** Garantir que todos os artefatos e a execução dos processos estejam em conformidade com os planos, procedimentos e padrões estabelecidos.
- d) **Gerência de Portfólio de Projetos.** Iniciar e manter projetos que sejam necessários, satisfatórios e sustentáveis, de forma atendam os objetivos estratégicos da organização.
- e) **Medição.** Coletar, armazenar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais.

8.4.1.3. Nível E – Parcialmente Definido

O nível E do Modelo MPS.BR, além dos processos dos níveis anteriores, é composto pelos processos “Avaliação e Melhoria do Processo Organizacional”, “Definição do Processo Organizacional” e “Gerência de Recursos Humanos”, com os seguintes propósitos:

- a) **Avaliação e Melhoria do Processo Organizacional.** Determinar o quanto os processos padrão da organização contribuem para alcançar os objetivos de negócio da organização e apoiar a organização a planejar, realizar e implantar melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos.
- b) **Definição do Processo Organizacional.** Estabelecer e manter um conjunto de ativos de processo organizacional e padrões do ambiente de trabalho aplicáveis às necessidades de negócio da organização.
- c) **Gerência de Recursos Humanos.** Prover a organização e os projetos com os recursos

- d) **Gerência de Reutilização.** Gerenciar os ciclos de vida dos ativos reutilizáveis.

Neste nível, o processo “Gerência de Projetos” sofre uma evolução, retratando seu novo propósito: gerenciar o projeto com base no processo definido para o projeto e nos planos integrados.

8.4.1.4. Nível D – Largamente Definido

O nível D do Modelo MPS.BR, além dos processos dos níveis anteriores, é composto pelos processos “Desenvolvimento de Requisitos”, “Integração do produto”, “Projeto e Construção do Produto”, “Validação” e “Verificação”, com os seguintes propósitos:

- a) **Desenvolvimento de Requisitos.** Definir os requisitos do cliente, do produto e dos componentes do produto.
- b) **Integração do produto.** Compor os componentes do produto, produzindo um produto integrado consistente com seu projeto, e demonstrar que os requisitos são satisfeitos para o ambiente alvo ou equivalente.
- c) **Projeto e Construção do Produto.** Projetar, desenvolver e implementar soluções para atender aos requisitos.
- d) **Validação.** Confirmar que um produto ou componente do produto atenderá ao seu uso pretendido quando colocado em ambiente de produção (ambiente para o qual foi desenvolvido).
- e) **Verificação.** Confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados.

8.4.1.5. Nível C – Definido

O nível C do Modelo MPS.BR, além dos processos dos níveis anteriores, é composto pelos processos “Desenvolvimento para Reutilização”, “Gerência de Decisões” e “Gerência de Riscos”, com os seguintes propósitos:

- a) **Desenvolvimento para Reutilização.** Identificar oportunidades de reutilização sistemática de ativos na organização e, se possível, estabelecer um programa de reutilização para desenvolver ativos a partir de engenharia de domínios de aplicação.
- b) **Gerência de Decisões.** Analisar possíveis decisões críticas usando um processo formal, com critérios estabelecidos, para avaliação das alternativas identificadas.
- c) **Gerência de Riscos.** Identificar, analisar, tratar, monitorar e reduzir continuamente os riscos em nível organizacional e de projetos.

8.4.1.6. Nível B – Gerenciado Quantitativamente

O nível B do Modelo MPS.BR é composto pelos processos dos níveis de maturidade anteriores (G ao C) e não possui processos específicos. Contudo, neste nível o processo de Gerência de Projetos sofre mais uma evolução, sendo acrescentados novos resultados para atender aos objetivos de gerenciamento quantitativo.

8.4.1.7. Nível A – Em Otimização

O nível A do Modelo MPS.BR é composto pelos processos dos níveis de maturidade anteriores (G ao B), atendendo integralmente todos os atributos de processos e não possui processos específicos.

O Processo e o Método de Avaliação do modelo MPS.BR, definidos em conformidade com a Norma Internacional ISO/IEC 15504-2:2003, verificam a maturidade da unidade organizacional na execução de seus processos e foram elaborados de forma que [SOFTEX 2009]:

- Permitem a avaliação objetiva dos processos de software de uma organização/unidade organizacional;
 - Permitem a atribuição de um nível de maturidade do MR-MPS com base no resultado da avaliação;
 - São aplicáveis a qualquer domínio na indústria de software;
 - São aplicáveis a organizações/unidades organizacionais de qualquer tamanho.
- O processo de avaliação descreve o conjunto de atividades e tarefas a serem realizadas para a organização atingir o seu propósito. Este processo possui quatro subprocessos, conforme ilustrado na figura Figura 8.7 a seguir, e obtém os seguintes resultados:
- São obtidos dados e informações que caracterizam os processos de software da organização/unidade organizacional;
 - É determinado o grau em que os resultados esperados são alcançados e os processos atingem o seu propósito;
 - É atribuído um nível de maturidade do MR-MPS à organização/unidade organizacional.

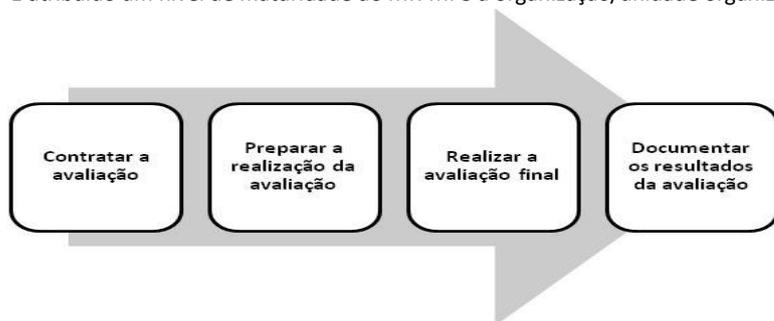


Figura 8.7. Subprocessos do Processo de Avaliação do MPS

O patrocinador pode ser um representante da alta gerência da unidade organizacional a ser avaliada, ou de outra organização que solicita a avaliação da unidade organizacional por uma terceira parte para fins de contrato. Para que uma avaliação seja conduzida com sucesso, é necessário:

- Comprometimento do patrocinador.** O comprometimento do patrocinador é essencial para assegurar que os objetivos da avaliação sejam atingidos. Este comprometimento também diz respeito aos recursos necessários, tempo e pessoal disponível para executar a avaliação.
- Motivação.** A atitude da gerência da unidade organizacional tem forte impacto nos resultados de uma avaliação. O responsável pela unidade organizacional deve motivar os participantes de forma aberta e construtiva. Deve, também, deixar claro a todos que o foco da avaliação é o processo e não o desempenho dos indivíduos que implementam o processo.
- Fornecimento de *feedback*.** O fornecimento de *feedback* e o estabelecimento de uma

avaliação, ajudam a assegurar que a avaliação seja significativa para a unidade organizacional.

- d) **Confidencialidade.** O respeito à confidencialidade das informações alcançadas durante a avaliação é essencial para que se obtenha as informações necessárias. Deve-se assegurar total confidencialidade aos participantes, tanto da equipe de avaliação quanto dos entrevistados. Da mesma forma, esse entendimento sobre a confidencialidade é essencial para que nenhum entrevistado se sinta ameaçado e todos se expressem livremente. A confidencialidade também abrange as informações contidas nos documentos apresentados pela unidade organizacional aos avaliadores.
- e) **Percepção dos benefícios.** Os membros da unidade organizacional devem perceber que a avaliação resultará em benefícios que os ajudarão direta ou indiretamente a realizar o seu trabalho.
- f) **Credibilidade.** O patrocinador, o gerente e os colaboradores da unidade organizacional devem acreditar que a avaliação chegará a um resultado representativo da organização/unidade organizacional. É importante que todas as partes confiem que os avaliadores têm a experiência e competência para realizar a avaliação, são imparciais e têm um entendimento adequado da unidade organizacional.

8.4.2.1. Prazo e Exigência de Pessoal

Uma avaliação seguindo o MA-MPS tem validade de três anos a contar da data em que a avaliação final foi concluída na unidade organizacional avaliada.

A equipe de avaliação é composta por membros internos representantes da unidade organizacional e externos à unidade organizacional, para garantir que a equipe tenha o conhecimento da unidade organizacional que está sendo avaliada (membros internos) e que não tenha interesse direto no resultado da avaliação (membros externos).

Os representantes são avaliadores com os mesmos deveres e direitos dos demais e contribuem com seu conhecimento da empresa para que toda a equipe entenda melhor a organização, seus processos e os artefatos apresentados; devem ter independência para desempenhar o seu papel de avaliador, não devem ser superiores hierárquicos dos colaboradores que serão entrevistados e não podem ter tido uma participação significativa nos projetos que serão avaliados. O avaliador líder deve garantir que sejam selecionados representantes adequados.

Os membros externos são:

- O avaliador líder da Instituição Avaliadora (IA) ou líder em formação indicado pela SOFTEX que esteja autorizado a conduzir avaliação nos níveis indicados pela SOFTEX;
- O(s) avaliador(es) adjunto(s) da IA⁹;
- O(s) avaliador(es) em formação indicado(s) pela SOFTEX, se for o caso.

8.4.2.2. Descrição do Método

Cada um dos quatro subprocessos do processo de avaliação é descrito pelo seu propósito e de suas atividades e tarefas, conforme descrito nas tabelas a seguir:

Tabela 8.4. Subprocesso Contratar a Avaliação [SOFTEX 2009]

Contratar a Avaliação		
Propósito	Atividade	Tarefas
Estabelecer um contrato para realização de uma avaliação, solicitada por uma organização/unidade organizacional que queira avaliar seus próprios processos ou os processos	1. Pesquisar Instituições Avaliadoras	1.1 Consultar site SOFTEX 1.2 Solicitar propostas para a avaliação
	2. Estabelecer contrato	2.1 Elaborar e enviar proposta 2.2 Selecionar Instituição Avaliadora 2.3 Formalizar contratação da Instituição Avaliadora

Tabela 8.5. Subprocesso Preparar a Realização da Avaliação [SOFTEX 2009]

Preparar a Realização da Avaliação		
Propósito	Atividade	Tarefas
Comunicar a contratação à SOFTEX e esta autorizar a realização da avaliação, planejar a avaliação, preparar a documentação necessária para a sua realização e fazer uma avaliação inicial que permita verificar se a unidade organizacional está pronta para a avaliação MR-MPS no nível de maturidade pretendido	1. Viabilizar a avaliação	1.1 Comunicar à SOFTEX a contratação da avaliação 1.2 Analisar a composição da equipe de avaliação e indicar o auditor da avaliação 1.3 Solicitar à unidade organizacional participação de avaliador em formação 1.4 Pagar taxa SOFTEX 1.5 Autorizar a realização da avaliação
	2. Planejar a avaliação	2.1 Enviar modelo do Plano de Avaliação à unidade organizacional 2.2 Planejar a avaliação inicial
	3. Preparar a avaliação	3.1 Enviar modelo da Planilha de Indicadores e Acordo de Confidencialidade à unidade organizacional 3.2 Preencher Planilha de Indicadores
	4. Conduzir a avaliação inicial	4.1 Assinar comprometimento com o Plano de Avaliação 4.2 Assinar o Acordo de Confidencialidade 4.3 Treinar equipe de avaliação para a avaliação inicial 4.4 Apresentar os processos da unidade organizacional 4.5 Verificar os indicadores de implementação 4.6 Analisar os dados da avaliação inicial 4.7 Enviar ao auditor a documentação da avaliação inicial 4.8 Auditar a avaliação Inicial 4.9 Realizar ajustes na documentação da avaliação inicial (se pertinente)
	5. Completar a preparação da avaliação	5.1 Completar Plano de Avaliação 5.2 Realizar ajustes (se pertinente) 5.3 Confirmar a realização da avaliação final

Tabela 8.6. Subprocessos Realizar a Avaliação Final [SOFTEX 2009]

Realizar a Avaliação Final		
Propósito	Atividade	Tarefas
<p>Treinar a equipe para a realização da avaliação final, conduzir a avaliação final, comunicar seus resultados à unidade organizacional avaliada e avaliar a execução do processo de avaliação na unidade organizacional</p>	<p>1. Conduzir a avaliação final</p>	<p>1.1 Realizar reunião de abertura 1.2 Assinar comprometimento com o Plano de Avaliação 1.3 Completar assinaturas do Acordo de Confidencialidade (se pertinente) 1.4 Treinar equipe para a avaliação final 1.5 Verificar evidências 1.6 Realizar entrevistas 1.7 Registrar afirmações na Planilha de Indicadores 1.8 Caracterizar o grau de implementação de cada resultado esperado do processo e de cada resultado esperado de atributo do processo em cada projeto 1.9 Caracterizar, inicialmente, o grau de implementação de cada resultado esperado do processo e de cada resultado esperado de atributo do processo na unidade organizacional. 1.10 Caracterizar, inicialmente, o grau de implementação de cada atributo do processo na unidade organizacional. 1.11 Caracterizar o grau de implementação, na unidade organizacional, de cada resultado esperado do processo, de cada resultado esperado de atributo do processo e de cada atributo do processo em reunião de consenso 1.12 Caracterizar o grau de implementação dos processos na unidade organizacional 1.13 Apresentar pontos fortes, pontos fracos e oportunidades de melhoria 1.14 Rever a caracterização e finalizar a redação dos pontos fortes, pontos fracos e oportunidades de melhoria (se pertinente) 1.15 Atribuir nível MR-MPS 1.16 Comunicar o resultado da avaliação ao patrocinador 1.17 Comunicar o resultado da avaliação aos colaboradores da unidade organizacional 1.18 Organizar ambiente de trabalho da avaliação</p>
	<p>2. Avaliar a execução do processo de avaliação</p>	<p>2.1 Avaliar a execução da avaliação pelo patrocinador 2.2 Avaliar a execução da avaliação pela equipe de avaliação 2.3 Avaliar a execução da avaliação pelo coordenador da IA 2.4 Avaliar a execução da avaliação pelo coordenador da IOGE (se pertinente) 2.5 Avaliar a execução da avaliação pela II (se pertinente) 2.6 Enviar avaliações do processo de avaliação para o auditor</p>

Tabela 8.7. Subprocesso Documentar os Resultados da Avaliação [SOFTEX 2009]

Documentar os Resultados da Avaliação		
Propósito	Atividade	Tarefas
Elaborar o Relatório da Avaliação, reunir a documentação da avaliação final e enviá-la ao auditor designado. Após a aprovação da documentação, o avaliador líder envia ao patrocinador o Relatório da Avaliação e comunica o resultado da avaliação à SOFTEX. O auditor envia a documentação da avaliação à SOFTEX, que insere os dados da avaliação em sua base de dados e divulga o resultado em seu site	1. Relatar resultados	1.1 Preparar o Relatório da Avaliação e o Resultado da Avaliação 1.2 Enviar ao auditor a documentação da avaliação final 1.3 Arquivar a documentação da avaliação final 1.4 Auditar a avaliação final 1.5 Realizar ajustes na documentação da avaliação final 1.6 Enviar Relatório da Avaliação ao patrocinador 1.7 Enviar Comunicação do Resultado da Avaliação à SOFTEX 1.8 Enviar documentação da avaliação à SOFTEX
	2. Registrar resultados	2.1 Inserir unidade organizacional no banco de dados SOFTEX 2.2 Divulgar no site SOFTEX (se pertinente) 2.3 Enviar o Acordo de Confidencialidade para a unidade organizacional com a declaração SOFTEX de avaliação de processos de software (placa de aço escovado) 2.4 Enviar o Acordo de Confidencialidade para a IA

CMMI x MPS.BR

O modelo de qualidade CMMI é reconhecido internacionalmente e se tornou uma referência no mercado. Empresas como a *Microsoft* já adotam o modelo como estratégia para exportação da mão-de-obra brasileira, buscando obter um diferencial competitivo.

Em contrapartida, o Modelo de Melhoria do Processo de Software Brasileiro é um modelo nacional. Pelo fato deste modelo apresentar mais estágios que o CMMI, permite uma implementação mais gradual e adequada às pequenas e médias empresas brasileiras. E, apesar de não ser um modelo internacional, está sendo difundido para outros países do Mercosul [KOSCIANKI 2007].

O modelo CMMI é proprietário e envolve um grande custo para realização das atividades do modelo para obter a certificação. Além disso, é necessário investir tempo; geralmente para se chegar aos níveis de maturidade mais altos leva-se em média de 4 a 8 anos [KOSCIANKI 2007].

Essas dificuldades contrastam com a realidade das empresas brasileiras que não podem realizar um investimento tão alto na obtenção da certificação. Logo, o modelo MPS.BR, com um custo mais acessível e com a exigência de um período de adequação inferior, supre essa necessidade das empresas brasileiras que desenvolvem software.

Contudo, com a implantação de qualquer um dos modelos citados acima, pode-se alcançar os seguintes benefícios:

- O desenvolvimento de software com qualidade, garantindo o cumprimento dos prazos e atendendo às necessidades do cliente, deixando-o mais satisfeito com o produto entregue pela empresa;
- Eliminação de inconsistências e redução de duplicidade;
- Utilização de terminologia comum e estilo consistente;
- Compatibilidade com a norma ISO/IEC 15504.

Todavia, uma vez que o modelo MPS.BR foi baseado em padrões internacionais de

breve comparação entre estes dois modelos: o MPS.BR e o CMMI (na representação por estágios).

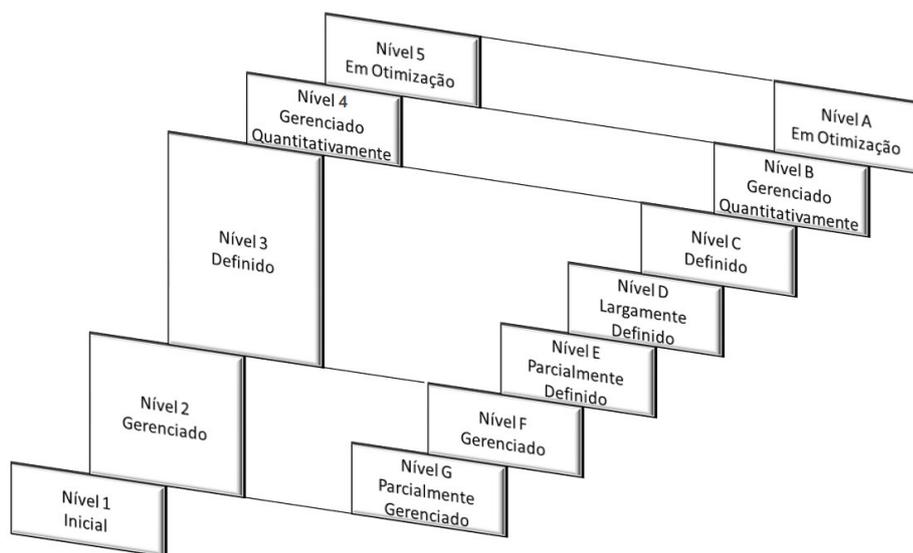


Figura 8.8. CMMI x MPS.BR

Não obstante, atualmente existem alguns processos que são implementados no modelo MPS.BR, mas que não possuem uma área de processo equivalente no CMMI, a exemplo do processo “Gerência de Portfólio de Projetos”. Isto se deve à nova atualização do MPS.BR (versão 2009). Logo, é possível concluir que poderá existir uma organização em conformidade com o MPS.BR, mas não estará em conformidade com o CMMI; o que não acontecia na versão anterior, onde prevalecia a equivalência entre ambos.

Exercícios

11. No modelo Capability Maturity Model Integration (CMMI), quantos e quais são os níveis de capacidade para a representação contínua?
12. No CMMI com representação em estágios, em qual nível de maturidade o desempenho dos processos é controlado usando estatísticas e outras técnicas quantitativas?
13. O gerenciamento de requisitos é uma das áreas-chave de que nível do modelo CMMI?
14. As empresas em conformidade com nível de maturidade 5 do CMMI (por estágio) possuem quais áreas de processo?
15. Qual o objetivo da representação contínua do CMMI e quais parâmetros são usados para ela?

16. Quais os pontos positivos e negativos do modelo CMMI?
17. Descreva como se dá o Processo de Avaliação para analisar se uma organização está em conformidade com o MA-MPS.
18. Quais os processos que devem ser implementados no nível A do MPS.BR?
19. Quais os processos que são implementados pelo MPS.BR que não são abordados pelo CMMI?
20. Indique as principais diferenças entre o modelo CMMI e o MPS.BR.

Sugestões de Leitura

1. **CMMI - Integração dos Modelos de Capacitação e Maturidade de Sistemas**
Autores: Couto, Ana Brasil
2. CMMI versus Modelagem BPMN
<http://www.scribd.com/doc/19077058/Modelagem-Das-Areas-de-Processo-Do-Cmmi-Usando-Bpm-Cmmi-e-SpemINFOCOMP2008>
3. MPS.BR - Melhoria do Processo de Software Brasileiro
<http://www.softex.br/mpsbr>

Tópicos de Pesquisa

1. Software Engineering Institute
<http://www.sei.cmu.edu/cmmi/>
2. SOFTEX - para Associação para Promoção da Excelência do Software Brasileiro
<http://www.softex.br/>
3. Qualidade de Software
<http://qualidadesoftware.org.br/>

Referências

- BETH, Mary; KONRAD, Mike; SHRUM, Sandy. **CMMI: Guidelines for Process Integration and Product Improvement**. Second Edition. Hardcover, 2006
- KNEUPER, Ralf. **CMMI: Improving Software and Systems Development Processes**. First Edition Usa: Rocknook, 2009.
- Itaborahy, A. , Radis, E. , Longhi, F. , Oliveira, K. M. e Figueiredo, R. M. C. "Aplicação do método SCAMPI para avaliação do processo de gerenciamento de projetos de software numa instituição financeira" (2005).
- Almeida, J. , Gracia, C. , Junior, F. e Dias, D. "Visão geral do Método de Avaliação Padrão CMMI para Melhoria de Processos – SCAMPI" (2007).
- SCAMPI. Members of the Assessment Method Integrated Team, "Standard CMMI Appraisal Method for Process Improvement (SCAMPI), version 1.1: Method Definition Document", Software Engineering Institute, Carnegie Mellon University, 2001.
- SEI - Software Engineering Institute. **CMMI® para Desenvolvimento – Versão 1.2**", Carnegie Mellon University, 2006
- KOSCIANKI, Andre; SOARES, Michel S. **Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. 2ª Edição. São Paulo: Novatec Editora, 2007.
- SOFTEX. MPS.BR - Melhoria de Processo do Software Brasileiro. www.softex.br. Acessado em agosto de 2009.

Capítulo

10

Implantação e Melhoria de Processos de Software

Jair da Silva Farias

O mercado de software tem evoluído exponencialmente adjunto à popularização dos computadores e dispositivos móveis, fato este, que deriva da globalização e da necessidade de uma economia mais competitiva, onde se busca um diferencial estratégico, ocasionando uma necessidade de processos que objetivem a qualidade dos produtos de software.

A qualidade dos processos para produção de software não garante, mas aumentam a probabilidade de que os produtos sejam de qualidade. Para se atingir níveis de maturidade e qualidade dos processos de software, organizações como o SEI, Softex e Motorola, criaram modelos que acompanham ou propõem práticas e processos para produção de software e redução dos defeitos.

Portanto é importante notar que a infraestrutura criada para realizar a Melhoria do Processo de Software (MPS) deverá desempenhar um papel significativo no sucesso ou fracasso de uma iniciativa de MPS. O valor que a infraestrutura traz a uma iniciativa de MPS, a compreensão das suas funções e responsabilidades, não pode ser subestimado.

Este capítulo aborda os modelos para implantação e melhoria de processos de software, dentre os modelos que serão abordados estão: o Seis Sigma que foi criado na década de 80 para reduzir o nível de defeitos na produção da Motorola. O IDEAL que foi criado pelo SEI para melhoria de processos organizacionais e o PRO2PI criado por Salviano, que foi baseado na norma ISO/IEC 15504 e propõe uma engenharia de processo dirigida por perfis de capacidade de processo.

Introdução a modelos para melhoria de processos de software

A primeira abordagem sistêmica de processos foi iniciada na década de 30 com Walter Shewhart [Shewhart 1980] com um trabalho em melhoria de processos com ênfase nos princípios do controle estatístico, sendo estes, refinados posteriormente na década de 80 em por de W. Edwards Deming [1986] e Joseph Juran [1997]. Entretanto, estes trabalhos eram focados na indústria de manufatura, que posteriormente seriam utilizados em fábricas de software.

Moreira [2008] descreve que na indústria de software, Watts Humphrey foi um dos primeiros estudiosos a aplicar o conceito de processo para a solução dos problemas de desenvolvimento de software. Em seu livro, [Humphrey 1989] afirma que *“o primeiro passo importante ao lidar com problemas de desenvolvimento de software é tratar esta atividade como um processo que pode ser controlado, medido e melhorado”*.

Em diversas fontes da literatura direcionadas a processos de software, [Moreira 2006] encontrou definições para o processo de software. Veja abaixo que em sua totalidade todas tem algo em comum:

- Um conjunto de atividades e resultados associados que produzem um produto de software [Sommerville 2006];
- Uma seqüência de passos executados para um determinado propósito [IEEE 2000];
- Um conjunto de atividades inter-relacionadas ou interativas que transforma entradas em saídas [ISO9000 2000];

Diante destas definições foi preciso modelar e aperfeiçoar um conjunto ou vários conjuntos de atividades, práticas e meios para a produção de software. Segundo [Moreira 2008] diversas pesquisas realizadas nos últimos anos têm apresentado a importância do uso sistemático e disciplinado de processos para que uma empresa de software possa obter sucesso. Este sucesso está relacionado a aspectos como, por exemplo: aumento de sua competitividade, capacidade para assumir maiores riscos, aumento da qualidade de seus produtos, ganhos com produtividade, menos custos e eliminação de re-trabalho. Há duas décadas, [Humphrey 1989] já afirmava que, para que empresas de software possam obter sucesso é preciso existir harmonia entre seus

Esta afirmação de Humphrey é legitimada por [Pressman 2002] que enfatiza que a falta de adoção de métodos, ferramentas e procedimentos no desenvolvimento de software têm alcançado números expressivos de projetos não concluídos, e projetos concluídos e que não atendem as necessidades do cliente.

Estudos e pesquisas tem concentrado a Engenharia de Software em uma subárea específica denominada de Melhoria de Processo de Software (*Software Process Improvement - SPI*), a MPS orienta que para desenvolver software de qualidade é preciso que os passos para seu desenvolvimento sejam acompanhados de atividades planejadas, gerenciadas, de modo a minimizar os custos e otimizar a realização das tarefas [Moreira 2008].

Pesquisadores [Habib et. Al, 2008] afirmam que “*qualquer melhoria de processo de software significante requer um investimento significativo, tempo e dinheiro*”. Então para que essas variáveis não sejam desperdiçadas é preciso um estudo de viabilidade e planejamento da mudança e da melhoria, por que *mudança não se faz da noite para o dia*.

Nas próximas seções serão detalhados os principais modelos que orientam a implantação e a melhoria dos processos de software, são eles: IDEAL, PRO2PI, Seis Sigma e o DMAIC como ferramenta de apoio ao Seis Sigma.

IDEAL

O IDEAL é um processo de melhoria de software criado na década de 90, que é usado para guiar o desenvolvimento de um plano estratégico integrado de melhoria a longo prazo, para o início e gestão de um programa de MPS. O objetivo desta seção e suas subseções é proporcionar ao leitor, uma descrição genérica de uma seqüência de passos recomendados para melhoria de processos de software baseada no modelo IDEAL.

O nome do modelo é formado pelo acrônimo das palavras (*Initiating, Diagnosing, Establishing, Acting, Learning*). O IDEAL é um modelo para programas de Melhoria de Processo de Software que foi desenvolvido pelo *Software Engineering Institute* (SEI) baseado no arcabouço de experiências de trabalhos de melhoria com o Governo Norte-Americano e outros clientes.

A proposta do modelo IDEAL está centrada na melhoria dos processos de

que são distribuídas em cinco fases: Iniciação (*Initiating*), Diagnóstico (*Diagnosing*), Estabelecimento (*Establishing*), Ação (*Acting*) e Aprendizagem (*Learning*) conforme ilustrado na Figura 1.

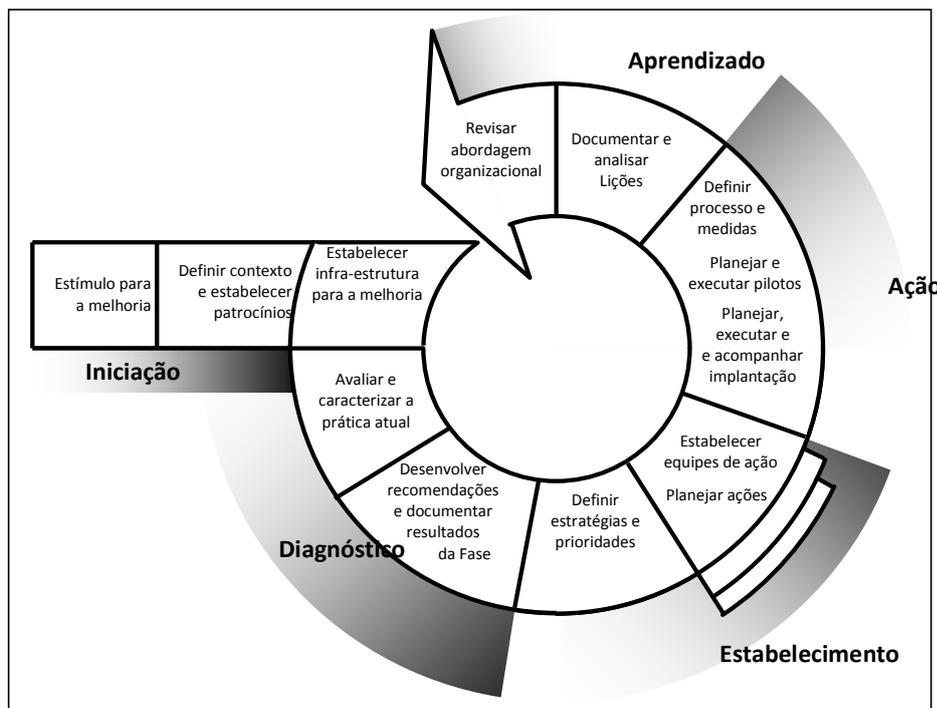


Figura 4 - Tradução do Ciclo do modelo IDEAL, Fonte: Moreira (2008) Apud McFeeley, 1996

Para aplicar o modelo IDEAL deve ser lembrado que existem dois componentes na atividade de melhoria de processo de software, o estratégico e o tático. Se o componente estratégico for baseado nas organizações empresariais, fornecerá a orientação e priorização das atividades táticas. A figura 2 mostra uma visão bidimensional da aplicação do modelo IDEAL.

Esta seção e suas derivadas destinam-se a abordar esses dois níveis operacionais dentro de um processo de iniciativa de melhoria, visto que são os níveis abordados pelo manual oficial do SEI, escrito por [McFeeley 2006], para a implementação do IDEAL:

- O nível estratégico, em que há processos que são a responsabilidade da gerência sênior.

- O nível tático, em que os processos são modificados, criado e executado por gerentes de linha e profissionais.

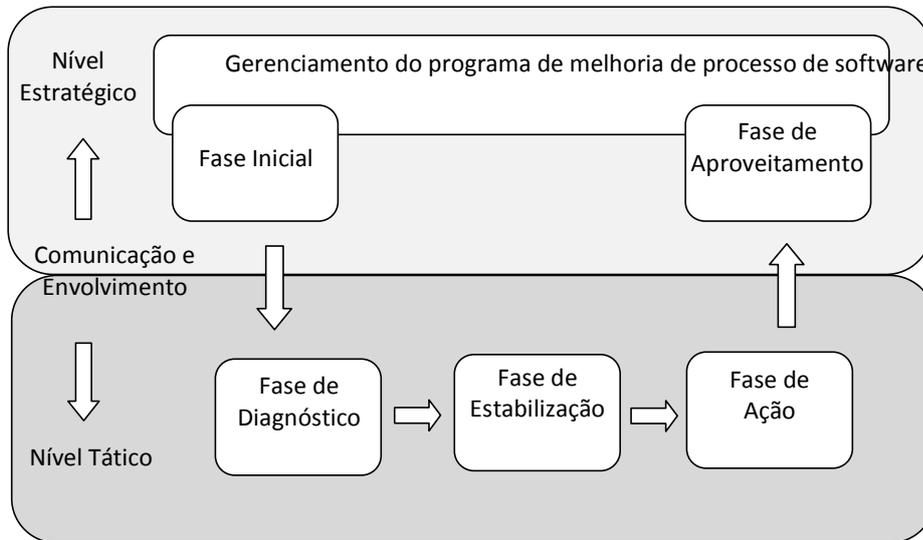


Figura 5 - Duas dimensões da atividade de melhoria de processos através do Ideal, Adaptado de McFeeley, 2006.

Na próxima seção serão explicadas detalhadamente, de acordo com [McFeeley 2006], as fases do IDEAL e a atividade de gerenciamento que é essencial para que a implementação do programa de melhoria obtenha maiores chances de sucesso.

▪ **Fases do IDEAL**

A base de todo modelo de melhoria é alicerçada em ciclos, visto que, a atividade de melhoria, seja de processos ou de qualquer setor organizacional é sempre contínua, esses ciclos são baseados em fases, e estas fases são compostas por processos, práticas e atividades.

Assim também acontece com o IDEAL. O modelo é composto por cinco fases e uma atividade gerencial, as fases são formadas de atividades-processos que devem ser seguidas para um maior alcance dos objetivos organizacionais, a atividade gerencial é essencial para o controle e evolução da MPS, aquelas e esta serão explicadas nas seis seções que procedem.

- **Fase inicial (Initiating)**

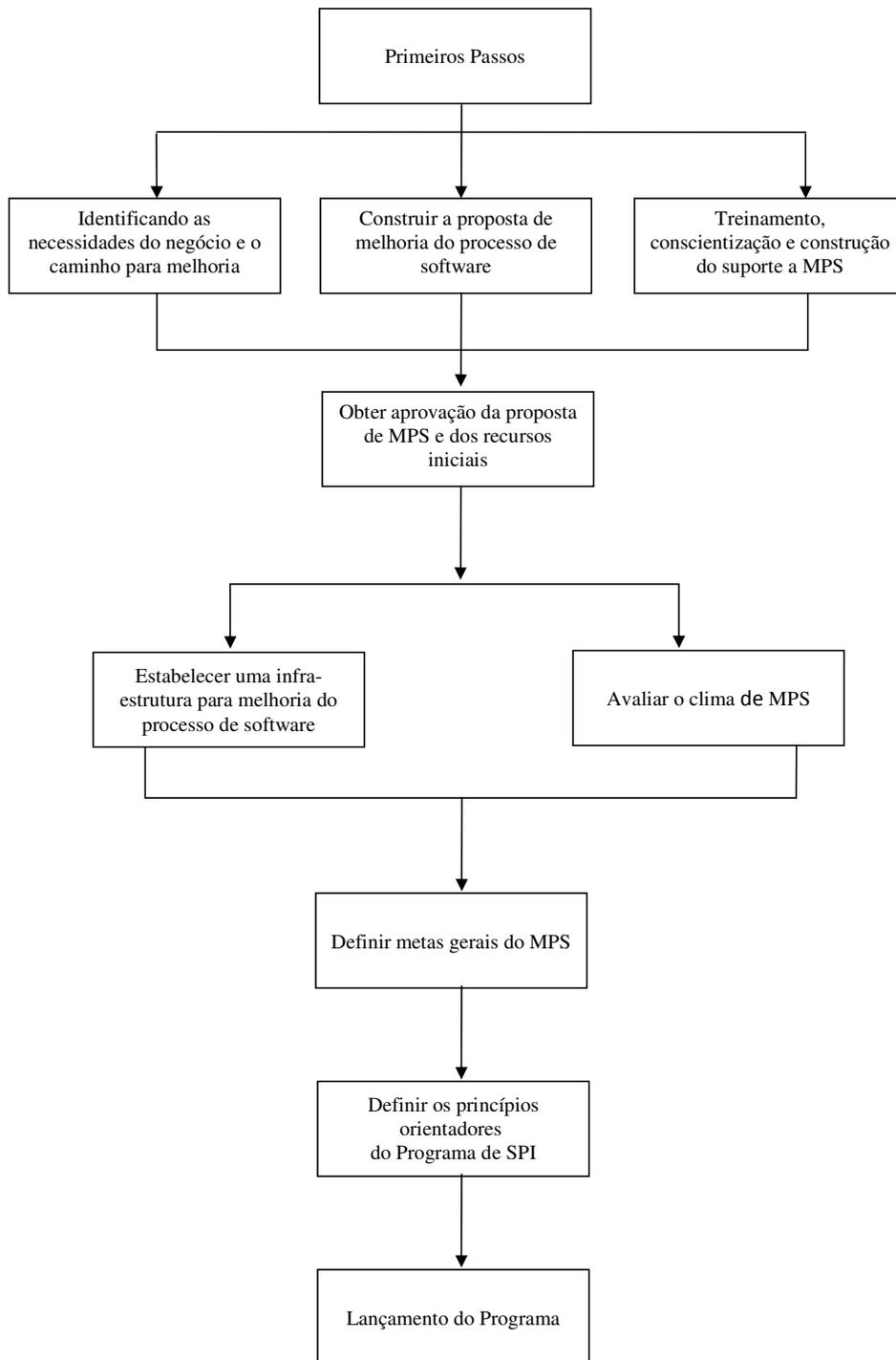
Este é o passo inicial para implementação do modelo IDEAL. Nesta fase a gerência sênior da organização compreende a necessidade de melhoria de processo de software (MPS), compromete-se com um programa de MPS, e define o contexto da MPS.

Esta etapa é similar à definição de um novo sistema. Um plano inicial de alto nível de MPS e um cronograma para as tarefas iniciais de MPS são desenvolvidos, e ainda, os principais elementos funcionais são definidos com uma chave de interfaces e os requisitos também são definidos e acordados. Este plano de alto nível vai orientar a organização até a conclusão da fase de estabelecimento, na qual um plano de ação para MPS será concluído. Normalmente, uma equipe é formada para explorar as questões e desenvolver uma proposta para MPS à gerência sênior. Após a aprovação da proposta da MPS, a infra-estrutura para o lançamento do programa de MPS será formada.

A organização precisa decidir como vai organizar os esforços de melhoria que serão envolvidos, tanto na prática e nos níveis de gestão, quanto no tempo das pessoas que serão afetadas. Além disso, é nesta fase que são formados os grupos para apoiar e facilitar o programa. Geralmente são formados dois grupos para o apoio a MPS, são eles:

- Grupo de Gerenciamento (*Management Steering Group - MSG*): responsável por vincular o programa de melhoria de processo à visão e missão da organização, representando dessa forma, o patrocinador. Este grupo é responsável por alocar e monitorar o progresso dos recursos, prover direcionamento e aplicar ações corretivas para o programa, quando necessário;
- Grupo de Processos de Engenharia de Software (*Software Engineering Process Group - SEPG*): indicado pela alta gerência para ser responsável por coordenar o programa de melhoria. Este grupo é responsável por promover, instruir e acompanhar as atividades, garantindo o bom andamento do programa dentro da organização.

A Figura 3 ilustra os dez processos sugeridos pelo guia de implantação do IDEAL para a primeira fase, após esta, seguiremos com a fase de diagnóstico da organização para dar continuidade a MPS.



- **Fase de diagnóstico (Diagnosing)**

O grupo de gerenciamento (*MSG*) deve compreender a base da organização e do processo atual de software para desenvolver um plano, que permita atingir o negócio na mudança específica do processo de software e nas metas da MPS da organização. As atividades realizadas na fase de diagnóstico vão fornecer essas informações para o planejamento e priorização da MPS.

É necessário para fornecer orientações claras para a melhoria de processos um plano estratégico de ação para a MPS. Através deste, diversas ações serão tomadas nos próximos anos, além disso, deverá fornecer, de forma clara e mensurável, as necessidades de negócio para a condução do programa de MPS, ligada ao plano de negócios da organização e da visão empresarial.

As linhas de base irão fornecer informação sobre como a organização atualmente realiza suas atividades de software. O conhecimento dos pontos fortes e oportunidades para melhoria é um pré-requisito essencial para a identificação e priorização de um eficaz e eficiente programa de MPS.

O resultado principal desta fase são as conclusões finais e relatório de recomendações, que é produzido como resultado das atividades de *baseline*¹⁰. Saídas secundárias podem ser revisões à visão da organização e do plano de negócios, um conjunto mínimo recomendado de *baselines* inclui:

- Organização de maturidade do processo de *baseline*.
- Descrição inicial do processo (mapa inicial do processo de software)
- Métricas de *baseline* (nível inicial do negócio e métricas de processo para medir o progresso).

Para cada *baseline*, muitos métodos eficazes de coleta de informação estão disponíveis. Para a *baseline* de maturidade do processo, um avaliador autorizado pode levar em conta a conduta da organização baseado no *Capability Maturity Model Integration* (CMMI) ou os colaboradores da própria organização pode ser treinado para avaliar o seu processo de maturidade. O MSG deve escolher o número e o tipo de *baseline* que melhor atingir os objetivos que fixou para que um relatório e

recomendações possam ser obtidas a partir de cada um. Manter uma dinâmica da melhoria de processos entre *baselining* e a implantação de MPS é muito importante.

A Figura 4 ilustra os seis processos sugeridos pelo guia de implantação do IDEAL para a fase de diagnóstico, após esta, seguiremos com a fase de estabilização da organização para dar continuidade a MPS.

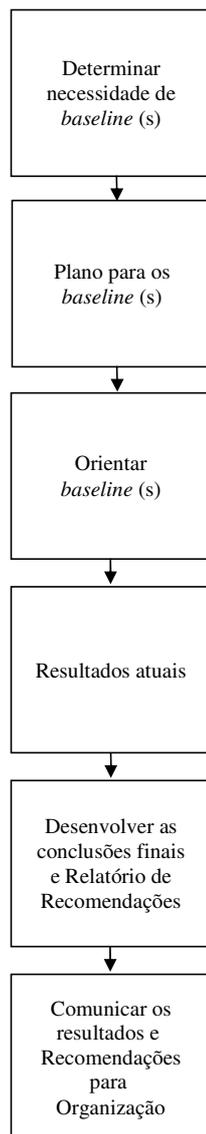


Figura 7 - Fluxo do Processo da fase de Diagnóstico, Adaptado de [McFeeley 2006]

- **Fase de estabilização (Diagnosing)**

Criar um plano de ação estratégico para a melhoria de processo de software (MPS) é um dos passos mais críticos e negligenciados da iniciativa de MPS. Por isso, é necessário que a equipe de gestão desenvolva ou atualize um plano de ação estratégico baseado na visão da organização, o plano de negócios, e as lições aprendidas dos esforços de melhoria passados, adjuntos aos resultados do *baselining* de esforços.

A fase de estabilização é repetida conforme necessário. Geralmente é desencadeado pela falta de um plano de ação para uma organização em seu primeiro ciclo, através do modelo IDEAL. Para aquelas organizações em um ciclo posterior, este passo pode ser desencadeado por uma necessidade de atualizar o plano anterior, objetivos, ou metas.

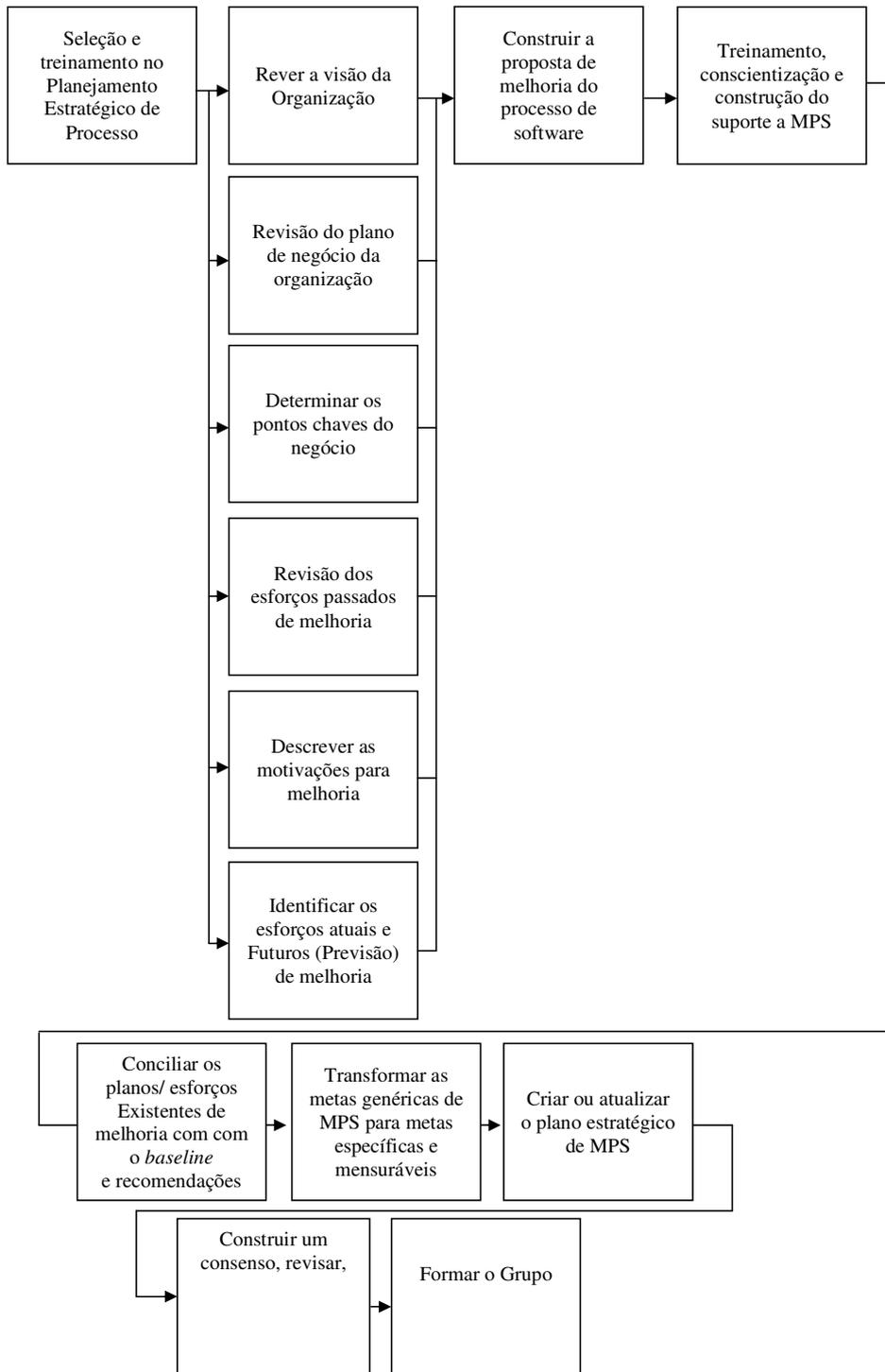
Nesta fase criar um plano de ação sólido é muito importante, as experiências mostram que, sem um planejamento cuidadoso, os esforços acabarão por falhar e haverá distorções, ou não correspondendo às expectativas escritas pelo alto gerenciamento. A razão que leva à necessidade de planos estratégicos bem elaborados, não é apenas identificar as melhorias, mas atender as necessidades críticas do negócio com a instalação dessas melhorias em toda a organização [McFeeley 2006].

A identificação das melhorias é muitas vezes a parte mais fácil. Fazer com que todos em toda a organização mudem a maneira como eles fazem as coisas é sempre a parte mais difícil de todo o esforço de melhoria.

O objetivo desta fase está centrado em desenvolver ou aperfeiçoar um plano estratégico de ação, que irá fornecer orientações e diretrizes para o programa de MPS, que terá duração de três a cinco anos, este é o tempo indicado pelo guia de implantação do IDEAL, para uma atualização do plano estratégico [McFeeley 2006]. A saída principal desta etapa é o plano de ação estratégica de MPS, saídas secundárias podem ser as revisões da visão da organização e do plano de negócios.

Além da produção de um plano estratégico de MPS é preciso integrá-lo com iniciativas já previstas ou em andamento de Gestão da Qualidade Total (*TQM*), com as conclusões e recomendações da *baseline* no plano de ação estratégico e alinhá-lo com o plano de negócio da organização, missão e visão. A Figura 5 ilustra os quatorze processos sugeridos pelo guia de implantação do IDEAL para a fase de estabilização

após esta, seguiremos com a fase de ação organizacional que é onde serão empregados os esforços planejados nas fases anteriores para dar continuidade a MPS.



- **Fase de ação (Acting)**

A fase de ação é a fase onde as melhorias são desenvolvidas e implantadas em toda a organização. Aqui as melhorias sugeridas pelos grupos de trabalho terão seu valor colocado a “prova”. O grupo de gestão (*MSG*) e o de processo de engenharia de software (*SEPG*) farão não só a gestão, mas também, o apoio ao desenvolvimento, controle, e implantação das melhorias.

A fase de ação une a missão do programa de MPS com a missão da organização no desenvolvimento de produtos. Este é o ponto culminante dos esforços de MPS. Para o planejamento e a introdução de melhorias, devem ser estudadas e avaliadas a estrutura atual da organização e as práticas utilizadas na criação dos produtos de software, para que elas sejam totalmente compreendidas e documentadas.

Também é importante um mecanismo ou métricas que identifiquem os efeitos da mudança em uma determinada área. Estes efeitos devem ser identificados o mais cedo possível para que eles possam ser tratados em tempo hábil. Para ajudar a compreender as práticas, é preciso se utilizar das técnicas disponíveis para modelar e avaliar as práticas atuais em “como estão”, e assim determinar a áreas de melhoria, e como os processos candidatos a melhoria devem ser examinados e avaliados.

Após essa avaliação e criação do estado atual dos processos, a organização precisa definir um "onde chegar" e escolher a solução adequada para atingir o estado desejado dos processos candidatos. Após esta avaliação e seleção, informar as decisões a serem tomadas para a seleção candidatos e a tecnologia a ser utilizada para a melhoria. A identificação de onde se quer chegar no estado do processo é muito importante para o sucesso global da fase de ação.

Esta fase do IDEAL é onde os Grupos de Trabalho Técnicos (*GTT*) desenvolvem melhorias específicas para processos específicos. Há duas abordagens básicas para concepção de soluções: foco na resolução de problemas específicos; incremento de um determinado processo.

Para esta fase, é essencial a utilização de projetos piloto para validar, refinar e testar os refinamentos das soluções para a MPS, esta fase pode necessitar de mais tempo que as demais por ser também uma fase experimentação. A Figura 6 ilustra os doze processos sugeridos pelo guia de implantação do IDEAL para a fase de ação após esta

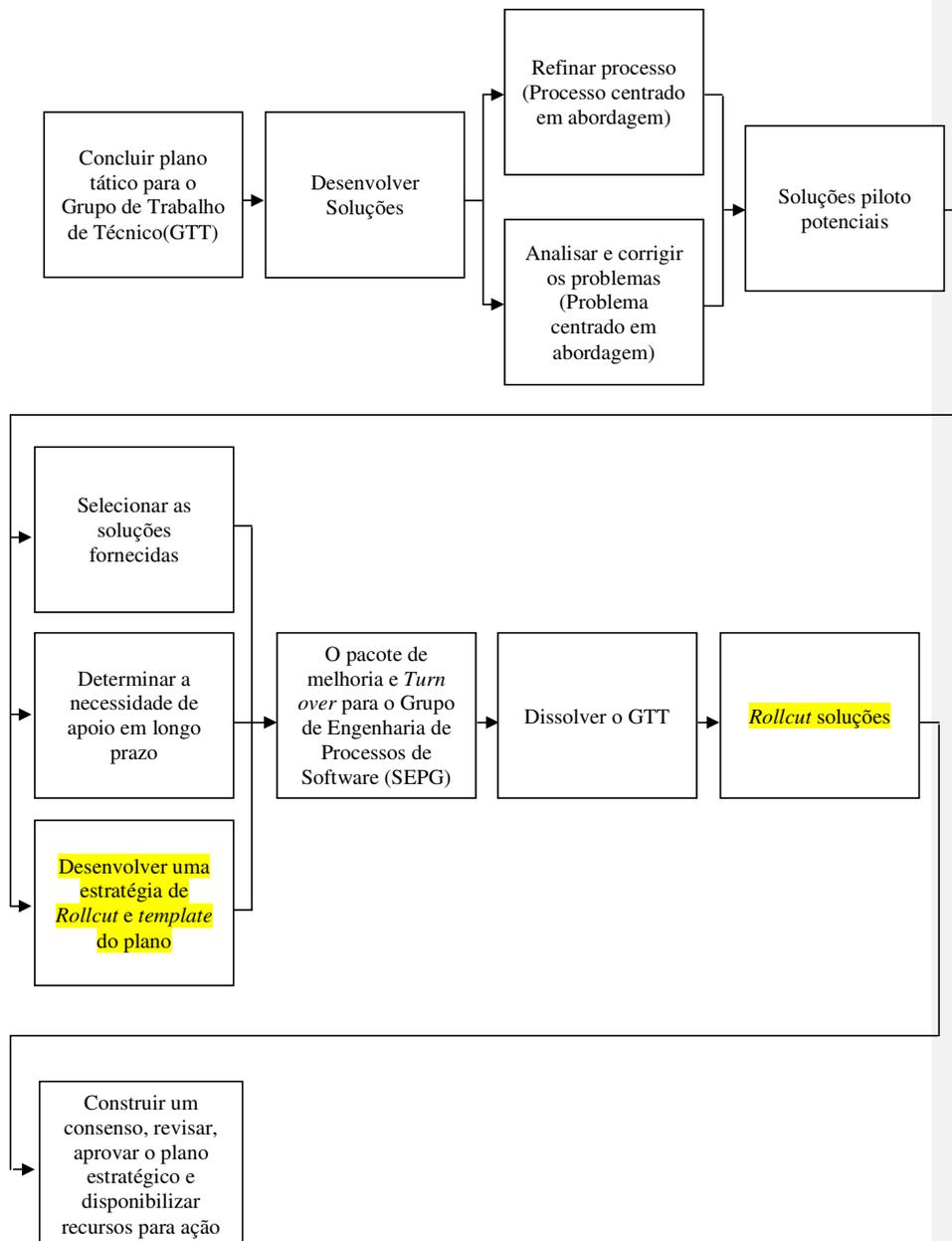


Figura 9 - Fluxo do processo da fase de ação, Adaptado de [McFeeley2006]

- **Fase de aproveitamento (Leveraging)**

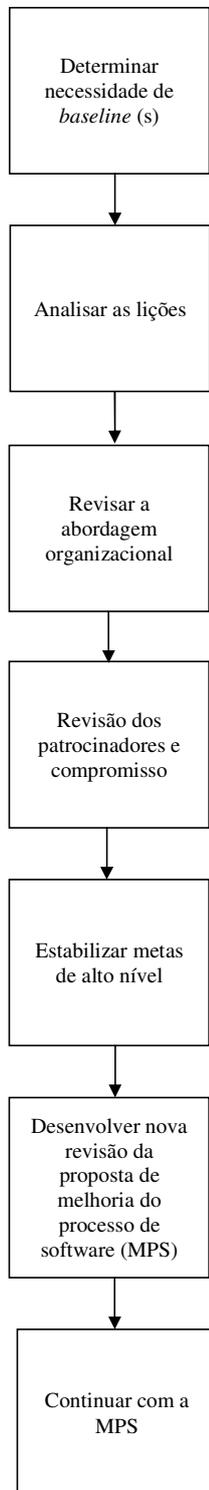
Agora que a organização completou um ciclo através do IDEAL, é necessário rever o que aconteceu durante esse ciclo e se organizar para o próximo ciclo, através do modelo. Ao invés de re-introduzir o IDEAL na fase de iniciação, esta fase vai retorna a fase de diagnóstico. Esta fase, além de preparação para o próximo ciclo através IDEAL, dá a oportunidade de ajuste a melhoria do processo de software (MPS) antes de iniciar o processo novamente.

Geralmente existem alguns “falsos começos” de melhoria em determinadas áreas da organização ou omissões e algumas atividades que foram planejadas para serem feitas mais que não ocorreram durante o primeiro ciclo do IDEAL. Uma vez que se tem documentado uma lista completa das lições aprendidas em cada uma das atividades de MPS, agora é preciso aplicá-las durante a fase de aproveitando a tornar o processo de MPS um trabalho mais eficiente e eficaz durante o próximo ciclo através do modelo IDEAL. Segundo [McFeeley 2006] algumas tarefas necessárias nesta fase são:

- Rever e analisar as lições aprendidas com as fases anteriores;
- Incorporar melhorias nos processos de MPS;
- Motivar a revisão das atividades de MPS;
- Analisar e avaliar as metas;
- Avaliar o patrocínio e empenho do envolvidos;
- Desenvolver um plano para proporcionar uma orientação contínua para o SPI programa.

Como visto nos tópicos acima, esta fase é dirigida a análise e revisão das tarefas que foram envolvidas no trabalho de MPS, através dessas análises e revisões é dado início a caracterização da melhoria contínua, baseada nas mudanças de melhoria dos processos e nas lições aprendidas com a MPS.

A Figura 7 ilustra os sete processos sugeridos pelo guia de implantação do IDEAL para a fase de aproveitamento, após esta, segue-se com a fase de gerenciamento do programa de melhoria do processo de software da organização para dar continuidade a MPS.



- **Fase de gerenciamento do programa de melhoria do processo de software (Manage)**

A melhoria do processo de software é uma iniciativa muito importante para uma organização. Para coordenar as diversas atividades que irão ocorrer no decurso de um programa de melhoria do processo de software (MPS) é necessária a previsão de uma efetiva infra-estrutura de apoio. Além disso, a infra-estrutura deve ser capaz de reagir de forma oportuna para as demandas do programa de MPS.

No início do programa de MPS, uma infra-estrutura de MPS inicial deve ter sido posta em prática para gerir as atividades da organização durante o programa. Esta fase de gerenciamento do programa de melhoria é o melhor momento para rever a forma como esta infra-estrutura tem realizado suas tarefas é após algum tempo criar o grupo comprometido com o apoio a MPS e a obtenção de patrocínios, avaliação do empenho de todos, completando as atividades *baselining* e o planejamento de ação para o próximo ciclo de IDEAL.

Com o programa de MPS em curso, uma infra-estrutura possivelmente como a ilustrada na Figura 8 deve ser desenvolvida e posta em prática. Esta infra-estrutura terá a responsabilidade de fornecer orientação para o programa de MPS. Na maioria dos casos, haverá três componentes para a organização: Grupo de Processos de Engenharia de Software (SEPG); Grupo de Gestão (MSG); Grupo de Trabalho Técnico (GTT).

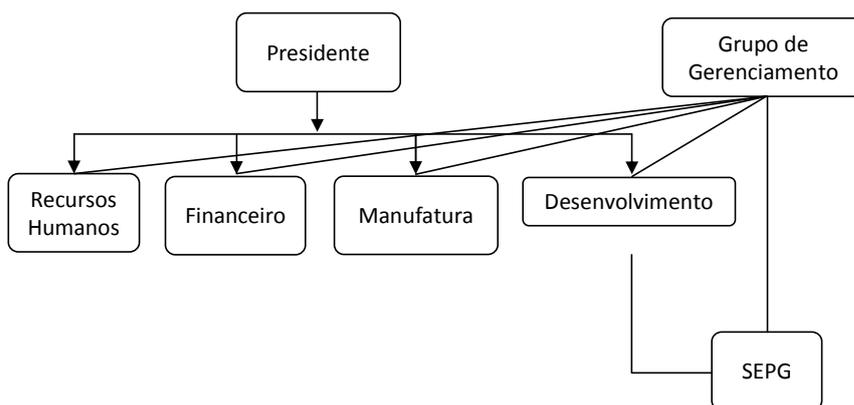


Figura 11 – Componentes típicos de uma infra-estrutura de MPS , Adaptado de [McFeeley2006]

Estes são os nomes genéricos e podem variar de organização a organização. Os componentes da infra-estrutura e sua relação com os outros são em grande parte determinados por fatores, como tamanho da organização e a diversidade geográfica.

Em [McFeeley 2006] são citadas algumas perguntas a responder sobre o desempenho da infra-estrutura que inicialmente foi posta em prática:

- A infra-estrutura está efetivamente ligada ao programa de MPS para a missão da organização e da visão organizacional?
- A infra-estrutura foi capaz de obter e alocar recursos suficientes para garantir conquistas oportunas?
- A infra-estrutura acompanhou corretamente o programa de MPS e forneceu orientação e correção necessárias?

As atividades de melhoria não irão ocorrer em um vácuo nem ocorrerão em série. Depois que o programa de MPS está em curso, haverá várias atividades de melhoria ocorrendo em diferentes unidades organizacionais. Como exemplo, podem haver grupos técnicos de trabalho (GTT) abordando gerenciamento de configuração, gerenciamento de requisitos, planejamento do projeto, e as análises comparativas, todos podem ocorrer simultaneamente. A infra-estrutura de apoio deve manter o controle de tudo isso e estar preparada para fornecer a necessária supervisão e orientação a todas as atividades do programa de MPS.

A infra-estrutura de apoio deve estar ciente de que os GTT's podem e provavelmente vão funcionar em paralelo. De acordo com [McFeeley 2006], a qualquer momento, o grupo de apoio deve estar preparado para:

- Oferecer suporte para uma tecnologia que está sendo introduzida;
- Formação e coordenação de recursos;
- Continuação, construção e fornecimento de patrocínio;
- Proporcionar conhecimento de planejamento;
- Avaliar o impacto organizacional;
- Mostrar as lições aprendidas.

implementar e gerir o modelo em uma organização, veja as sugestões de leitura no final desse capítulo, no tópico IDEAL, lá você vai encontrar como adquirir o manual gratuito e completo do modelo.

PRO2PI

O aumento nos estudos voltados ao auxílio das atividades de produção software tem gerado uma diversidade de métodos, ferramentas, práticas, processos e metodologias para o desenvolvimento de software, entretanto estes são acompanhados de estudos de melhoria no decorrer do tempo.

Salviano (2006) cita que a melhoria de processo de software tem apontado na prática ser uma abordagem eficaz e eficiente para a necessária melhoria das organizações que produzem software. A comunidade tem relatado vários casos de sucesso, como, por exemplo, [Herbsleb et al. 1994, DACS 1999 e Card 2002].

O PRO2PI¹¹, criado por Salviano, surgiu baseado em “*Uma Proposta Orientada a Perfis de Capacidade de Processo para Evolução da Melhoria de Processo de Software*”, foi fruto resultante da junção dos estudos em Melhoria de Processos de Software, Modelos de Capacidade de Processo, Gerações de Arquiteturas de Modelos de Capacidade de Processo e Engenharia de Processos dirigida por perfis de capacidade.

Os estudos de Salviano apontam para uma grande quantidade de normas na literatura relativas às abordagens para melhoria de processo, como, por exemplo, IDEAL (McFeeley 1996), ISO/IEC 15504 [ISO/IEC TR 15504-7 1998, ISO/IEC 15504-4 2004], problemas e metas [Porter e Sakry 2002] e as orientações para a melhoria de [O’Toole 2000], que utilizam como referência um modelo de processo que sistematiza e representa as melhores práticas, definem uma medição para avaliação da capacidade dos processos e provê um roteiro racional para a melhoria dos processos.

Exemplos de modelos mais utilizados são a Norma ISO/IEC 12207 (1998¹²), a ISO/IEC 15504 que é também conhecida como *SPICE (Software Process Improvement and Capability Determination)* (ISO/IEC 15504 1998), o CMMI (*Capability Maturity Model Integration*) [Chrissis et al. 2003], e a aplicação para software da ISO 9000,

¹¹ PRO2PI: *PROcess capability PROfile to Process Improvement*, onde o 2 representa ao mesmo tempo a

principalmente a versão 2000 com o par coerente 9001 e 9004 (ISO 9001 2000, ISO 9004 2000).

Essas normas, metodologias, modelos e abordagens estão bem difundidas na comunidade e descritas em várias publicações [Rocha et al. 2001], inclusive com considerações sobre o relacionamento entre os modelos citados [Sheard 2001].

Na próxima seção o processo base de formação do PRO2PI será iniciado, baseado na engenharia de processo dirigida por perfis de capacidade e seus fundamentos. É importante observar a Figura 1 que descreve os princípios do processo de definição de um PRO2PI baseado nas necessidades organizacionais.

- **Engenharia de processo dirigida por perfis de capacidade e seus fundamentos**

Para que haja um processo de descoberta de oportunidades para evolução na área de melhoria de processo, é preciso estudar a engenharia de processo dirigida por níveis de capacidade de processo, nesse estudo é possível refletir sobre propostas de mudança de melhoria dos processos de software.

A definição para perfil de capacidade de processo aponta para um modelo que representa um processo segundo o aspecto de capacidade de processo. O processo de uma empresa deve ser representado por um perfil de capacidade de processo, que é uma abstração do processo, segundo o aspecto capacidade de processo. O par consistente formado por perfil de capacidade de processo e processo é ilustrado na Figura 1, onde um processo é uma parte do mundo (M0) e é representado, segundo o aspecto capacidade de processo, pelo modelo perfil de capacidade de processo no espaço de modelagem (M1) [Salviano 2006].

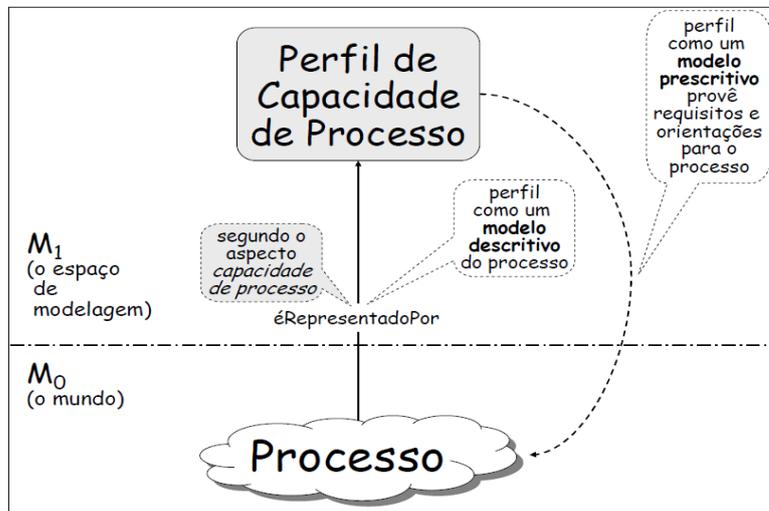


Figura 12 - Perfil de capacidade de processo e processo, Fonte: (Salviano 2006)

De acordo com [Salviano 2006] uma forma simples de entender a relação entre o perfil e o processo é com o seguinte questionamento: Se o perfil de capacidade de processo representado por um nível de maturidade do modelo CMMI/DEV tivesse sido definido para o processo de uma determinada unidade organizacional, em um determinado momento, qual seria este nível de maturidade de forma a representar o processo atual ou o processo alvo para uma melhoria alinhada ao contexto e objetivos estratégicos da organização.

A melhoria de processo deve ser evoluída para uma engenharia que trate esse par consistente (perfil e processo) no centro dessa engenharia. É proposto então, como uma evolução da atual melhoria de processo de software baseada em modelos de maturidade, uma Engenharia de Processo Dirigida por Perfil de Capacidade de Processo (*Process Capability Profile Driven Process Engineering - PCDE*) aplicada a software.

Na próxima seção a definição da abordagem PRO2PI apresentada detalhadamente, é importante a observação das ilustrações, pois estas modelam a forma de funcionamento da aplicação do PRO2PI possibilitando um melhor entendimento.

▪ **O PRO2PI**

As propriedades do PRO2PI foram definidas a partir de seis fases, onde foram

resumo das fases de formação de um PRO2PI está ilustrado na Figura 4 e podemos chamá-lo de “processo de montagem do PRO2PI”.

A Figura 10 exemplifica o modelo base para o ciclo de melhoria PRO2PI que é composto por melhores de práticas dos modelos de maturidade, práticas de desenvolvimento de software e modelos de gestão.

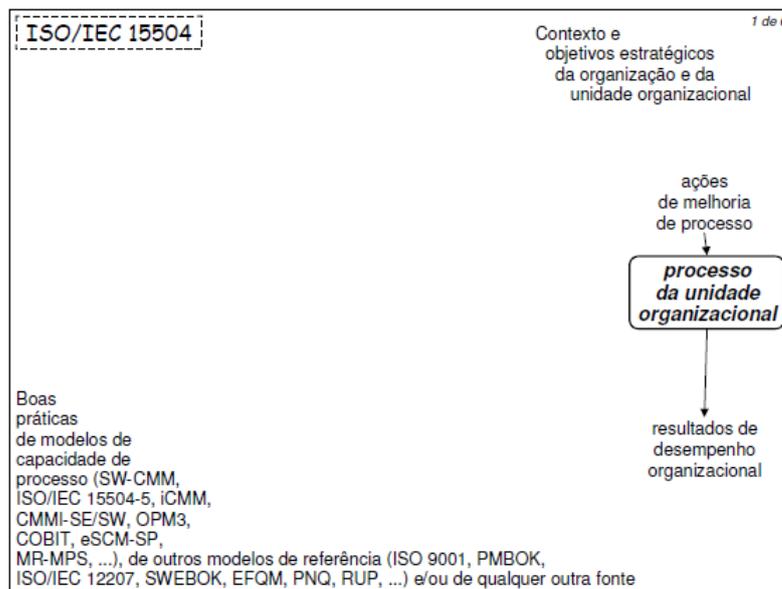


Figura 13 - Diagrama da utilização corrente da melhoria de processo, Fonte (Salviano, 2006).

A fase 2 do processo de formação do PRO2PI que é representada através da função de definição ou atualização de PRO2PI “defineP”, acontece a partir do momento em que uma organização seleciona elementos de um ou mais modelos de referências e define um perfil de capacidade de processo que representa os elementos selecionados desses modelos e de qualquer outra fonte. Um ciclo de melhoria de processo é realizado objetivando evoluir os processos para atender a esse perfil alinhado com o contexto e objetivos estratégicos da organização.

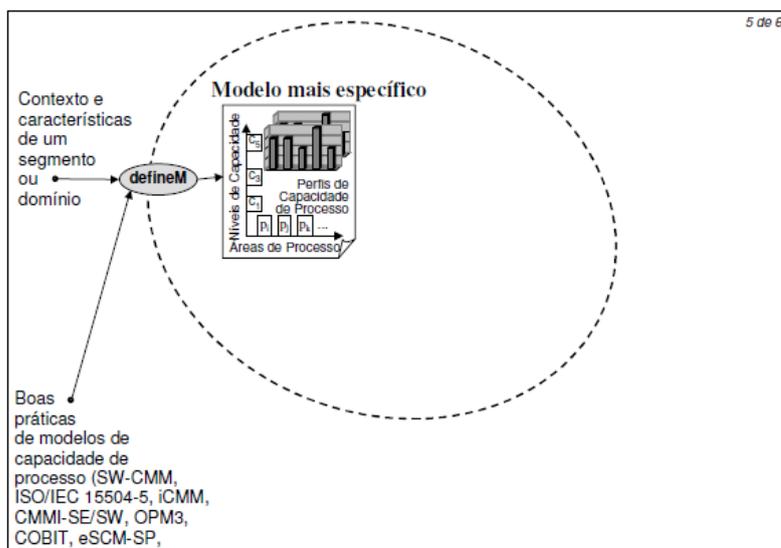
O perfil pode conter boas práticas de referência selecionadas dos modelos mais genéricos existentes e de outras fontes, de forma a representar orientações relevantes à organização. O perfil pode ser alterado a qualquer momento em função de novas percepções, alterações do contexto ou dos objetivos estratégicos e dos resultados da

utilização da versão corrente do perfil. Esse uso é representado pela função “defineP” (define, ou atualiza, perfil de capacidade de processo) da Figura 3, [Salviano 2006].

Já a fase 3 é perpassada pela função de utilização do PRO2PI, o uso desse perfil é utilizado para orientar as ações de melhoria e, é representado pela função “usaP” (usa perfil de capacidade de processo) na Figura 4. Neste caso as ações de melhoria devem ser satisfatórias para que o processo resultante atenda a todos os requisitos representados no perfil de capacidade de processo.

A fase 4 que conta com a função de avaliação de capacidade de processo em relação a um PRO2PI. O processo da organização pode ser examinado com uma avaliação de capacidade de processo em relação ao perfil de capacidade de processo. Esse exame é representado pela função “avaliaPr” (avalia capacidade de processo em relação a um PRO2PI) na Figura 4. Os resultados de capacidade de processo gerados por essa avaliação podem ser utilizados como referências adicionais para a definição, ou alteração do perfil de capacidade de processo [Salviano 2006].

O ciclo com a função de definição de modelos de capacidade de processo mais específicos com a abordagem PRO2PI é apresentado na Figura 3. A partir do contexto de negócio de um segmento, como, por exemplo, o segmento Metalúrgico, ou de um domínio, como, por exemplo, engenharia de testes, pode ser definido um modelo mais específico. Esse modelo pode ser composto por áreas de processo ou perfis de capacidade de processo relativos um segmento ou domínio.



A abordagem PRO2PI é caracterizada pela busca contínua ao alinhamento do perfil de capacidade de processo como uma representação, segundo o aspecto de capacidade de processo, do processo da unidade organizacional. Essa relação está descrita na Figura 1.

Em concordância com a Figura 3, a melhoria de processo com o PRO2PI é realizada com a definição e utilização de um PRO2PI que pode ou não conter boas práticas selecionadas dos modelos de capacidade de processo, modelos com outras finalidades, modelos mais específicos de um segmento ou domínio ou de qualquer outra fonte, que se adaptem as necessidades estratégicas da organização.

É importante salientar que a abordagem PRO2PI é baseada no ciclo de utilização das funções (“defineM”, “usaP”, “avaliaPr”, “defineP”) que giram em torno do PRO2PI e do modelo mais específico descrito anteriormente.

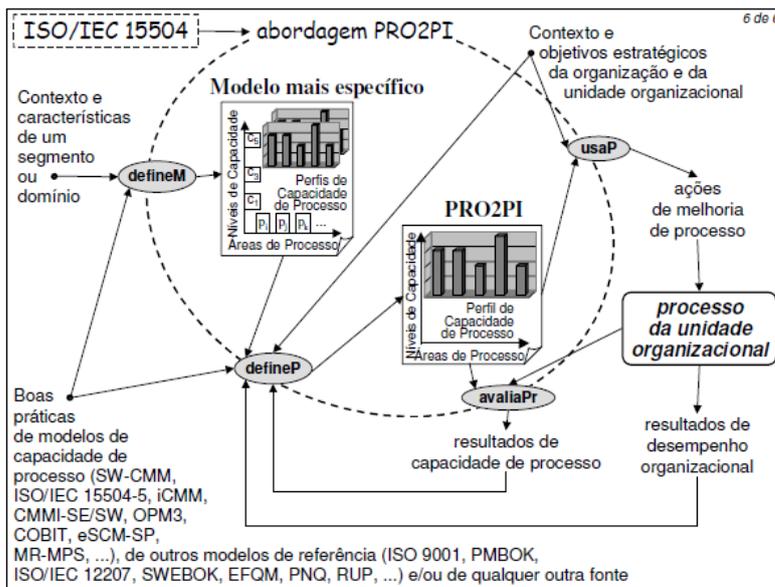


Figura 15 - Diagrama da abordagem PRO2PI para modelos e melhoria de processo, Fonte: (Salviano, 2006)

A Figura 5 demonstra uma analogia à utilização da abordagem PRO2PI ilustrada na Figura 4 com a utilização da engenharia de requisitos no desenvolvimento de software, que podemos denominar de desenvolvimento de software orientado por requisitos.

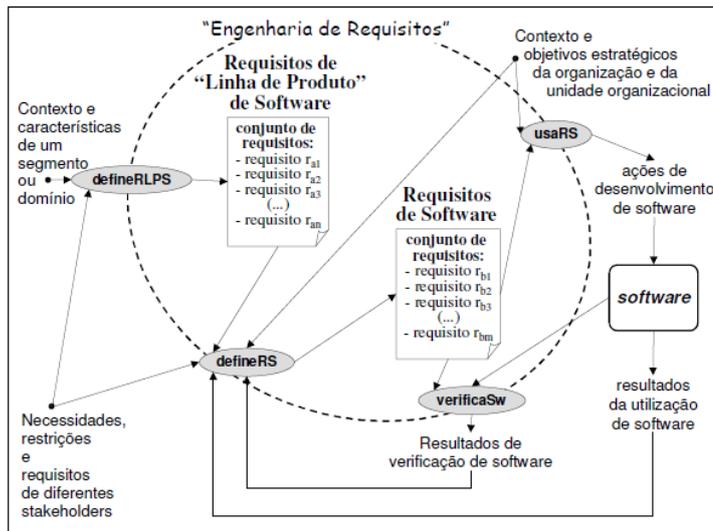


Figura 16 - Diagrama, análogo ao de PRO2PI, do desenvolvimento de software, Fonte: (Salviano, 2006)

Na Figura 6, a função “defineRS” (define requisitos) é utilizada para definir, ou atualizar, o conjunto de requisitos, a função “usaRS” (usa requisitos de software) será utilizada para guiar o desenvolvimento do software, de forma que, seja dirigida pelos requisitos. A função “verificaSw” (testa software) será utilizada para buscar problemas no software e com isto sugerir o quanto o software desenvolvido está em concordância com os requisitos. A função “defineRLPS” (define requisitos de linha de produto de software) representa o desenvolvimento de requisitos para uma linha de produto de software, de tal forma que esses requisitos sejam um sistema, sendo que os requisitos foram baseados em diferentes conjuntos de requisitos, de diferentes stakeholders. Esta forma de desenvolvimento de requisitos é análoga à forma de desenvolvimento de modelos mais específicos representada na Figura 4 [Salviano 2006].

As próximas seções deste capítulo referentes ao PRO2PI identificam um conjunto de quatro elementos que descrevem a abordagem e define cada elemento desse conjunto. Esses elementos são propriedades de um PRO2PI, um modelo de PRO2PI,

- **PRO2PI-PROP: Propriedades de PRO2PI**

De acordo com Salviano (2006) para ser útil e efetivo como orientação para a melhoria de uma organização, um PCP deve ter, em um grau suficiente, pelo menos as propriedades de ser relevante, oportuno, viável, representativo e específico em um determinado momento e com uma determinada previsão de investimento em ações de melhoria, ser sistêmico e dinâmico, e ser rastreável a modelos relevantes, conforme ilustrado na Figura 9.

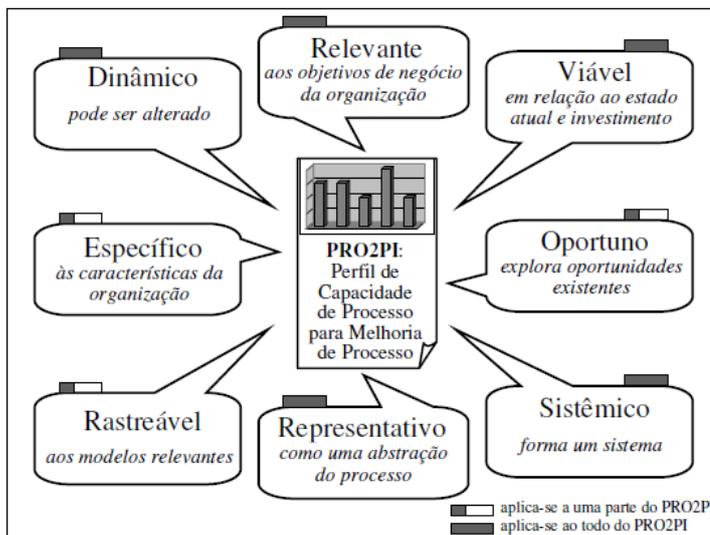


Figura 8 - Propriedades de PRO2PI, Fonte: (Salviano, 2006)

A Tabela 1, que é baseada na descrição de Salviano das propriedades de um PRO2PI, explica detalhadamente cada propriedade e as descreve com exemplos, após esta seguiremos para a seção do elemento modelo de PRO2PI (*PRO2PI-MODEL*).

Tabela 1 – Propriedades de um PRO2PI, Adaptado de Salviano 2006

Propriedade	Descrição
Relevante	Para ser relevante um PCP deve representar um estado do processo da organização que esteja alinhado com a estratégia de negócio da organização, e com isto represente melhorias importantes para a organização.
Viável	Um PCP deve representar um estado de processo que requeira uma quantidade de esforço e de recursos para seu atendimento, a partir do estado atual, que seja viável de ser disponibilizado. Para isto uma análise deve ser feita. É necessário a utilização de uma medição para esta quantidade que pode ser baseada na unidade de melhoria

Oportuno	Um PCP deve considerar, e apontar oportunidades disponíveis. Por exemplo, mesmo que a área de configuração não esteja com prioridade alta, mas existam recursos disponíveis a baixo custo, pode ser viável incluir devido à oportunidade.
Sistêmico	Um PCP deve representar um estado do processo que seja um sistema. Desta forma o processo representado por esse estado funcionará na organização, gerando os resultados necessários. Como um sistema, esse estado será ao mesmo tempo um situação auto sustentável e uma etapa para a melhoria.
Representativo	Um PCP deve representar uma abstração do processo, segundo o aspecto de capacidade de processo. Esta abstração deve ser completa, ou seja, representar todo o processo, nos termos e elementos de PCP. Todas as características representadas no PCP têm que estar sendo realizadas no processo e o processo realiza apenas o que está representado no PCP.
Rastreável	Como os modelos genéricos mais utilizados são consolidações de melhores práticas utilizadas por várias organizações, pode ser importante manter uma rastreabilidade do PCP com elementos relevantes, como, por exemplo, áreas de processo e níveis de maturidade, dos modelos relevantes para a organização.
Específico	Como é importante manter um relacionamento um para um entre um PCP e o estado do processo, e cada organização tem sua especificidade, é fundamental que o PCP tenha elementos específicos para a organização.
Dinâmico	Como os vários fatores que influenciam o que deveria ser o estado do processo podem mudar, é importante que os PCPs possam ser ajustados. Também como o conhecimento sobre esses mesmos fatores tende a aumentar, os PCPs podem ser ajustados para refletir esse melhor conhecimento. Portanto eles devem ser dinâmicos.

- **PRO2PI-MODEL: Modelo de PRO2PI**

O modelo de PRO2PI, denominado de *PRO2PI-MODEL* foi definido por Salviano (2006), para atender aos seguintes requisitos:

- Unificar os elementos e estruturas dos modelos de capacidade de processo considerados mais relevantes, especificamente dos modelos iCMM v2.0, CMMI-SE/SW v1.1, ISO/IEC 15504-5:2006 e MR-MPS v1.0, e com isto permitir a representação de praticamente qualquer elemento desses modelos em um PRO2PI;
- Buscar representar em um PRO2PI os elementos de outros modelos de capacidade de processo, incluindo PMI OPM3, ITsqc eSCM-SP, ITGI COBIT, SMMM, KMMM, PMMM, UMM, TMM, OOSPICE, SPICE4SPACE,

- Permitir também a representação em um PRO2PI de elementos selecionados de outros modelos de referência, que não sejam modelos de capacidade de processo, especificamente dos modelos PMBOK e ISO 9001, e incluindo ISO/IEC 12207 Amd2, IEEE SWEBOK 2004, EFQM, FPNQ PNQ e RUP.
- É especificado em uma referência conceitual adequada na linha de MDE;
- Minimizar a quantidade de tipos de elementos de outros modelos.

Salviano (2006) definiu o PRO2PI-MODEL direcionado à unificação e nas extensões de dois frameworks para modelos de capacidade de processo: o framework da ISO/IEC 15504 [ISO/IEC 15504-2 2003] e o do CMMI. A Figura X relaciona os principais elementos da estrutura dos modelos CMMI-SE/SW v1.1 e ISO/IEC 15504-5:2006 e indica como esses elementos são considerados na estrutura de PRO2PI-MODEL.

CMMI-SE/SW v1.1	PRO2PI-MODEL	ISO/IEC 15504-5 :2006
	(Grupo de Práticas)	
Área de Processo	Área de Processo	Processo
Propósito	Propósito	Propósito
Objetivo	Objetivo	"Propósito"
"Prática Específica"	Resultado	Resultado
Prática Específica	Prática Base	Prática Base
Produto Típico	Artefato	Produto de Trabalho
" "	Recurso	" "
	(Grupo de Práticas)	
Nível de Capacidade	Nível de Capacidade	Nível de Capacidade
Objetivo Genérico	Propósito	Propósito
"Prática Genérica"	Objetivo	Atributo de Processo
Prática Genérica	Resultado	Resultado
Sub-prática	Prática Base	Prática Genérica
" "	Artefato	Produto Genérico
" "	Recurso	Recurso

Figura 17 - Elementos da estrutura do CMMI e 15504-5 em PRO2PI-MODEL, Fonte: [Salviano, 2006]

Além disso, elementos de outros modelos, como ISO 9001, SWEBOK, PMBOK, OPM3 e SW-CMM, podem também ser modelados e associados ao PRO2PI-MODEL. Com isto é possível definir o perfil de capacidade de processo baseando-se em elementos de múltiplos modelos.

- **PRO2PI-MEAS: Medições para PRO2PI**

As medições de PRO2PI, denominadas de PRO2PI-MEAS (PRO2PI Measurements) são um conjunto de medições relacionadas a PRO2PI definidas por [Salviano 2006], estas medições podem servir para medir a viabilidade de um PRO2PI, complexidade e efetividade do investimento a ser realizado na MPS.

Segundo [Salviano 2006] o PRO2PI-MEAS utiliza o modelo de informação de medição definido na norma [ISO/IEC 15939 2002] e utilizado no PSM [McGarry et al. 2002] e nas áreas de processo de medição e análise dos modelos CMMI-SE/SW e ISO/IEC 15504-5. Para tanto são definidos:

- Uma medida derivada da complexidade de uma melhoria de processo representada por um PRO2PI (*Derived Measure Process Improvement Complexity* DM-PIC) em relação à situação atual de uma unidade organizacional, em termos de Unidades de Melhoria de Processo (*Unit for Process Improvement Complexity* U-PIC);
- Uma medida derivada da efetividade de investimento para melhoria de processo em uma unidade organizacional (*Derived Measure for Investment Effectiveness for Process Improvement* DM-IEPI), em termos de Unidades de Eficiência de Investimento para Melhoria de Processo (*Unit for Investment Effectiveness for Process Improvement* U-IEPI);
- O produto de informação denominado Viabilidade de PRO2PI, em termos de uma interpretação de DM-PIC e DM-IEPI.

A Figura 9 ilustra o modelo PRO2PI de medição, segundo a estrutura definida na Norma ISO/IEC 15939.

Em tese [Salviano 2006] explica que o modelo de análise da viabilidade é uma função da complexidade de um PRO2PI e da efetividade do investimento de uma unidade organizacional, e tem valor normalizado da viabilidade igual a 1, quando a complexidade da melhoria modelada por PRO2PI é compatível com a capacidade da unidade organizacional. O valor de viabilidade é maior que 1, quando a melhoria é maior que a capacidade, e menor que 1, quando a melhoria é menor que a capacidade. Um PRO2PI é considerável viável para uma unidade organizacional quando o valor

20% para cima ou para baixo. Para informações completas não só do modelo de medição, mas do PRO2PI por completo veja as sugestões de leitura no final capítulo.

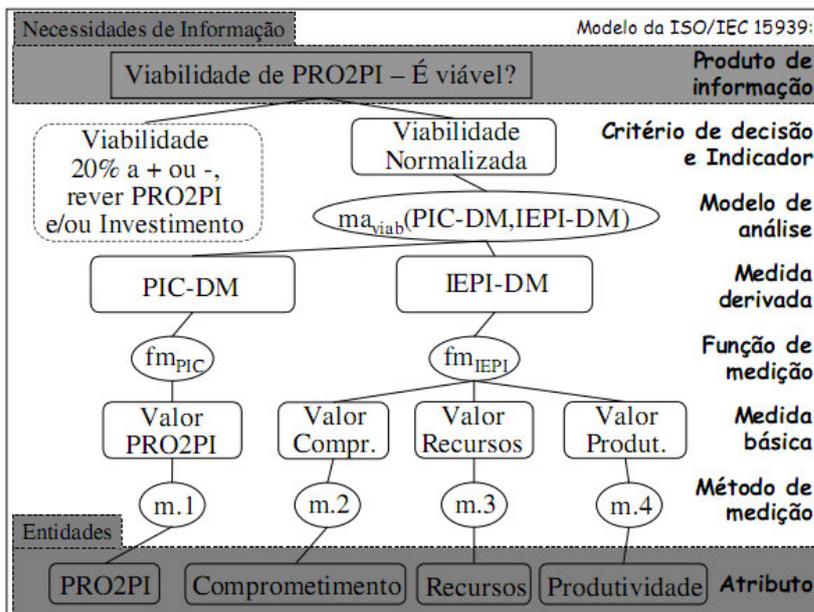


Figura 18 - Produto de informação Viabilidade de PRO2PI, Fonte: (Salviano, 2006)

- **PRO2PI-CYCLE: Processo para ciclo de melhoria**

Um programa de MPS baseado em modelos é realizado com a utilização de abordagens. Entre elas podem ser destacadas a abordagem IDEAL, citada na seção 1.1 deste capítulo, o ciclo de melhoria da ISO/IEC 15504 e a abordagem AMP1. O IDEAL foi definida para arquitetura estagiada fixa, e ciclo 15504-4 e AMP1 são definidas para arquitetura estagiada fechada.

O processo para ciclo de melhoria com PRO2PI utiliza como referência as três abordagens citadas. Um ciclo de melhoria com PRO2PI pode entendido como composto por fases correspondentes às fases das abordagens citadas com o acréscimo de uma atividade para definição e utilização de PRO2PI que pode ser utilizada em qualquer fase [Salviano 2006].

O programa é iniciado com a decisão e comprometimento da organização em realizar a melhoria. As atividades desse ciclo são alinhadas com o contexto e os objetivos estratégicos da organização e podem utilizar experiências e resultados de

melhoria porque o objetivo é a melhoria do negócio da organização, por meio da melhoria do processo de produção software.

A referência principal para a melhoria é o perfil de capacidade de processo, cuja definição e utilização é encapsulada em “define e utiliza PRO2PI” na Figura 10 que ilustra as seis fases e outros elementos do processo PRO2PI-CYCLE. O objetivo principal das atividades de definição e utilização de PRO2PI é definir e utilizar PRO2PI. O produto de entrada e de saída é o perfil de capacidade de processo. As atividades, que podem ser consideradas como uma implementação das práticas base definidas para o processo de estabelecimento de processo. Os objetivos de PRO2PI-CYCLE incluem [Salviano 2006]:

- Identificar e analisar os objetivos, estratégia, contexto e/ou qualquer outro aspecto relevante de negócio da unidade organizacional e da organização, para subsidiar e orientar a definição dos objetivos de melhoria;
- Identificar os objetivos da melhoria, incluindo objetivos mais específicos para o próximo ciclo de melhoria e objetivos mais gerais para o programa de melhoria, sempre alinhados aos objetivos, estratégia, contexto e/ou qualquer outro aspecto relevante de negócio identificados;
- Estabelecer critérios de qualidade para avaliar e melhorar um perfil de capacidade de processo

Assim é finalizada a seção de PRO2PI, para informações com maior nível de detalhe, veja a seção de sugestões de leitura no final deste capítulo. A próxima seção apresenta o Seis Sigma, que neste capítulo será abordado com um método estatístico para a melhoria do processo de software.

Seis Sigma

Esta seção irá abordar fatores relevantes sobre o Seis Sigma, PDCA e DMAIC que é uma ferramenta utilizada no Seis Sigma, baseada no PDCA, ela possibilita a melhoria de processos de software através de ciclos de atividades de melhoria.

O Seis Sigma é um método estatístico que é representado pela 18ª letra do alfabeto grego, o sigma (σ), que é também o símbolo de desvio padrão na estatística. Em [Wang 2008] o Seis Sigma é definido como uma abordagem que melhora a

qualidade através da análise de dados estatísticos. Nos últimos anos tem havido um aumento significativo em sua utilização.

Em pesquisa feita por [Moreira 2008] ela afirma que Seis Sigma é uma metodologia de melhoria de processo criada por Bill Smith para a Motorola em 1986 [Eckes 2001] com o objetivo de ajudar a empresa a reduzir custos e a melhorar a qualidade dos produtos através de ferramentas analíticas de medição e controle. Inicialmente o foco do Seis Sigma era melhorar os processos de manufatura, com o passar do tempo esta abordagem ganhou maturidade tornando-se amplamente utilizada por organizações de diversas áreas para melhoria de processos organizacionais.

“Seis Sigma é a inflexível e rigorosa busca da redução da variação em todos os processos críticos para alcançar melhorias contínuas e quânticas que impactam os índices de uma organização e aumentam a satisfação e lealdade dos clientes. É uma iniciativa organizacional projetada para criar processos de manufatura, serviço ou administrativo. A ferramenta de melhoria empregada na implantação dos projetos Seis Sigma é o DMAIC: acrônimo que representa: Definir-Medir-Analisar-Implementar-Controlar” [Rasis 2002]. Esta ferramenta será definida com mais detalhes na seção 1.5.2.

Comment [j212]: refazer

Existem várias definições para Seis Sigma, cada uma varia conforme a atividade onde está sendo aplicada a metodologia. *“Seis Sigma se refere a um processo em que o intervalo entre a média de um processo de medição da qualidade e o mais próximo da especificação limite é pelo menos, seis vezes o desvio padrão do processo”* [Wang 2008].

Comment [j213]: refazer

Six Sigma possibilita uma capacidade de processo que deve gerar apenas 3,4 defeitos por milhão de oportunidades (DPMO), apresentando 99,99966% de perfeição [Donegan 2005]. Com isso, pode-se inferir que, atingir o nível de Seis Sigma é um processo lento que exige muito planejamento e comprometimento com a qualidade do produto, fato este, que é medido através do histórico das variações dos defeitos ocorridos na produção e pode ser acompanhado pelo DMAIC.

A Tabela X demonstra uma relação numérica de níveis Sigma de acordo com o percentual de perfeição e DPMO do projeto, fazendo uma relação com o tempo desperdiçado causado pelos defeitos existentes.

Tabela x - Níveis Sigma (Fonte: Donegan, 2005)

Nível Sigma	Percentual Correto (%)	DPMO	Tempo Desperdiçado por Século
3	93,3193	66.807	3 ½ meses
4	99,3790	6.210	2 ½ dias
5	99,9767	233	30 minutos
6	99,99966	3,4	6 segundos

Segundo [Scatolin 2005] não se pode aceitar a ilusão de que Seis Sigma é a solução dos problemas para toda empresa. Deve-se fazer uma análise crítica e verificar se a metodologia é a mais adequada a depender do momento em que a empresa está.

O objetivo do Seis Sigma é suprir as necessidades de uma empresa em melhorar seus processos de forma contínua e sustentável. Através de um forte foco na capacitação e treinamento de seus colaboradores, as empresas que implementam esta metodologia têm a finalidade de diminuir o desperdício com defeitos e possíveis re-trabalhos e um aumento agressivo nos lucros, e com isso, proporcionar uma evolução contínua dos seus processos internos, incentivando o crescimento e melhorando o aproveitamento dos seus funcionários.

Portanto, a estratégia de melhorar o desempenho de processos, o aproveitamento de recursos materiais e o atendimento ao cliente, acompanhados por elevados investimentos no treinamento dos colaboradores e incentivando a criatividade, faz do Seis Sigma uma metodologia que consegue promover o atendimento dos objetivos da empresa que o adota, desde que, uma análise crítica entre custo e benefício seja realizada previamente.

Entretanto, Seis Sigma não é apenas uma simples medida de defeitos. Ela é uma abordagem holística para melhoria dos negócios que abrange: filosofia, medida de desempenho, metodologia para melhoria, e um conjunto de ferramentas. De acordo com Siviý et al. 2008], por causa de suas diversas dimensões adaptáveis as necessidades das organizações, o Seis Sigma tanto pode servir como um modelo de governança empresarial, através da melhoria de processos da alta administração, como um mecanismo técnico de melhoria, através da coleta de métricas de defeitos na produção de um determinado produto.

Por ser a base para formação do DMAIC, a próxima seção abordará o ciclo PDCA que é a ferramenta mais importante e mais utilizada para MPS é nela que foram

baseadas a definição da abordagem IDEAL [McFeeley 1996] e o ciclo de melhoria da ISO/IEC 15504.

- **PDCA**

Inicialmente o ciclo PDCA foi idealizado por Walter Shewart na década de 30, mas o mesmo só viria a ser disseminado largamente na indústria por [Deming 1986]. O nome do modelo PDCA é derivado do acrônimo das palavras (*Plan-Do-Check-Action*, Planejamento-Execução-Verificação-Ação) o modelo é visto na literatura como o precursor no conceito de melhoria de processos [Moreira 2008, Salviano 2006].

O ciclo PDCA é dividido em quatro fases, na primeira fase é feito um planejamento das ações a serem tomadas, estabelecimento de metas e fixação das atividades necessárias para atingir as metas. Na segunda fase são realizadas as atividades fixadas na primeira fase. Já na terceira fase é feita uma análise e verificação dos resultados atingidos. Finalmente, a quarta fase envolve correções de metas e se for necessário, são tomadas ações corretivas ou de melhoria se necessário [Salviano 2006].

- **DMAIC**

O DMAIC é a ferramenta do Six Sigma mais utilizada na melhoria de processos de software (MPS), ela aborda as fases e as atividades necessárias dentro do ciclo de melhoria utilizado para atingir as metas organizacionais.

Em pesquisa realizada por [Moreira 2008] é citado que o modelo DMAIC foi desenvolvido inicialmente pela Motorola como o modelo MAIC (*Measure, Analyse, Improve e Contro* - Medir, Analisar, Melhorar e Controlar) como evolução do ciclo PDCA e depois adotado pela GE como DMAIC, em que D (*Define*) significa a fase de definição. Esse método passou a ser a base operacional da ruptura do Seis Sigma para essas empresas, sendo fundamental para o sucesso que alcançaram [Harry e Schroeder, 1998, Pande et al. 1998, Eckes 2001].

Para uma definição enxuta e clara do ciclo, observe a Figura 11, para uma definição mais detalhada, leia as próximas seções baseadas em pesquisa feita por Morreira [2008] do ciclo DMAIC.

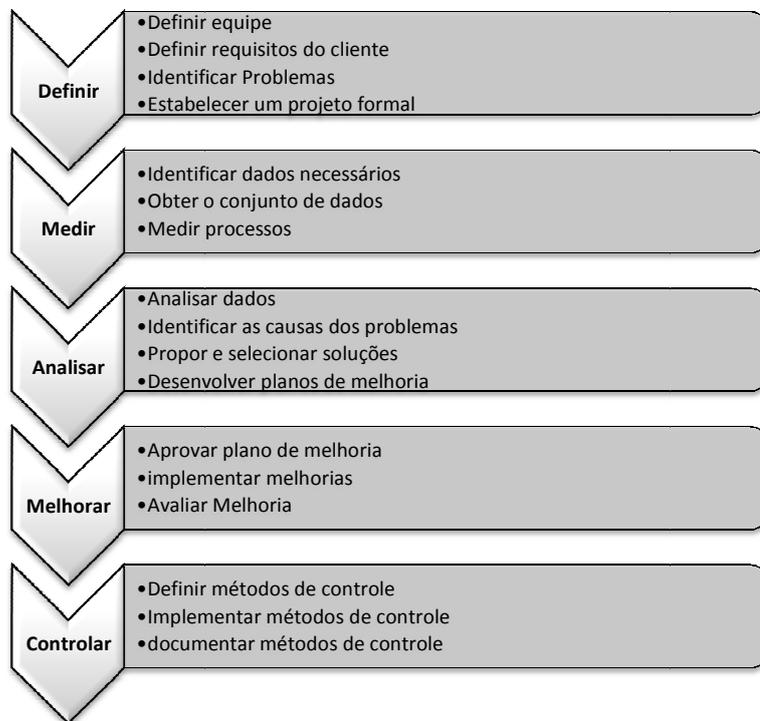


Figura 19 - Fases do ciclo DMAIC, Adaptado de [Siviy et al. , 2008]

- **Definir**

Essa fase de definição serve como plataforma para a equipe organizar-se, determinar seus papéis e responsabilidades, estabelecer metas e marcos (*milestones*) e rever passos do processo. Os pontos-chave a serem definidos são os requisitos do cliente, escopo do projeto, priorização de causa e escopo, e o planejamento do projeto. Cada um desses passos pode ser alinhado com o cliente e é essencial apreciar e entender essa ligação com o cliente antes e durante esse estágio do modelo.

As ferramentas utilizadas nesta fase incluem dados dos clientes, análise custo benefício, desenho dos macro-processos prioritários.

- **Medição**

A fase de medição apresenta os objetivos de confirmar e quantificar o problema; identificar variáveis importantes de entrada no processo; medir os passos do processo

atual; se necessário, revisar o problema; definir os resultados esperados e exibir as variações usando Diagrama de Pareto, histogramas, *run charts*.

A fase Medição é um importante passo da abordagem, pois os seus resultados são utilizados para a tomada de decisões baseadas em fatos. É também durante esta fase que é calculado o desempenho atual do processo (capacidade Sigma atual do processo).

- **Análise**

A fase Análise tem como objetivo principal analisar os dados coletados na fase Medição através ferramentas de análise para identificar as causas primárias dos problemas e propor soluções para os mesmos. Pode ser realizado um *brainstorming* durante esta fase, a fim de determinar as melhorias de maior impacto nos requisitos do cliente, levando em consideração os riscos associados.

Nesta fase é utilizada a análise de dados exploratória e descritiva para ajudar a entender os dados. Ferramentas estatísticas como teste de hipóteses, análise de variância e regressão são utilizadas para apoiar esta fase.

- **Melhoria**

Na fase melhoria o planejamento e as análises realizadas são executados. As decisões baseadas em fatos foram realizadas na fase de análise utilizando métricas da fase de medição, e agora as melhorias do processo resultante poderão ser implementadas, após serem aprovadas.

A solução proposta na fase anterior é avaliada e validada nesta fase e, se conveniente, é aplicada em larga escala na organização. Métodos como Delineamento de Experimentos (DOE) e gráficos estatísticos são empregados para validar a melhoria e o desempenho do processo (sigma) é recalculado.

- **Controle**

Esta fase é projetada para garantir que os ganhos conseguidos nas fases anteriores não sejam perdidos, medindo as melhorias e garantindo que sejam sustentadas. Para tanto, são elaborados procedimentos para medição e controle do processo (Controle estatístico do processo – CEP). Estes procedimentos são validados e documentados pela equipe do programa de melhoria. Neste momento é validado o

desempenho e o retorno financeiro do projeto junto com os patrocinadores e a equipe. Ferramentas como gráficos de controle são utilizados nesta fase do ciclo.

Considerações Finais

Este capítulo apresentou de forma detalhada os principais modelos para implantação e melhoria do processo de software, o IDEAL, PRO2PI, fundamentado na norma 15504, e o Seis Sigma.

Diante do arcabouço de melhoria de processo de software, fica claro que o mesmo é de uma complexidade, que muitas vezes se torna até subjetiva devido à quantidade de variáveis que envolve a melhoria de processos de software. É preciso que esta atividade passe por uma fase de planejamento e estudo detalhada, como todas as abordagens aqui citadas propõem, nenhuma delas garante que o retorno do investimento será imediato, por isso, é preciso muita cautela, um planejamento estratégico e uma equipe de apoio bem formada e treinada, para que seja possível dar início ao processo de melhoria.

Exercícios

- Descreva o que é melhoria de processo de software e qual a sua importância para organização.
- Descreva o que é IDEAL e qual o objetivo de implementá-lo.
- Quantas e quais são as fases do modelo IDEAL? Descreva cada uma delas.
- Segundo o modelo Ideal, quais os componentes típicos de uma infra-estrutura de MPS?
- Quais as propriedades de PRO2PI? Descreva cada uma delas.
- Quais são as funções utilizadas em um ciclo de PRO2PI?
- O que é Seis Sigma e qual o seu objetivo principal?
- Explique o que é PDCA e quais as suas fases.
- Explique o que é DMAIC e quais as suas fases. Descreva cada uma delas.
- Descreva um conjunto de práticas que possa unir as práticas do IDEAL, PRO2PI e Seis Sigma, para MPS.

Sugestões de leitura

Para entender melhor o que é Melhoria de processo de software lei a norma **ISO/IEC 15504-4/2004**.

Para um estudo detalhado sobre o PRO2PI leia tese de doutorado de Clênio Salviano, “Uma Proposta Orientada a Perfis de Capacidade de Processo para Evolução da Melhoria de Processo de Software”. Disponível em: <http://libdigi.unicamp.br/document/?code=vtls000380495>

Para um estudo detalhado sobre IDEAL leia o guia oficial de implantação produzido pelo SEI, “IDEAL - A User's Guide for Software process Improvement”. Disponível em: <http://www.sei.cmu.edu/library/abstracts/reports/96hb001.cfm>

Tópicos de pesquisa

Visto que a tarefa de MPS é realizada por pessoas os **aspectos humanos** envolvidos na MPS devem ser estudados com maior detalhe.

Aspectos psicológicos e sociais da MPS que podem influenciar da eficiência e eficácia de um programa de MPS.

Como a tarefa de MPS é complexa e muitas vezes subjetiva, estudar um **conjunto de Métricas eficazes para MPS** é uma tarefa delicada e que exige muitos anos de pesquisa.

O comprometimento dos Recursos Humanos com a MPS, ainda deve ser muito pesquisado.

Novas normas e padrões de MPS surgirão com o advento das Metodologias Ágeis, visto que a necessidade de Time-to-Marketing diminui.

MPS X Metodologias ágeis, ainda existem poucos estudos relacionados a esse tema.

Referências

Chrissis, M. Beth, Konrad, Mike and Shrum, Sandy (2003) "CMMI: Guidelines for Process Integration and Product Improvement", Addison-Wesley Pub Co, 2003.

- Data & Analysis Center for Software, A Business Case for Software Process Improvement Revised - Measuring Return on Investment from Software Engineering and Management, A DACS State-of-the-Art Report, Contract Number SP0700-98-4000, Prepared By Thomas McGibbon, 30 September 1999. (disponível em <http://www.dacs.dtic.mil/techs/roispi2/>, acessado mais em 23/08/2009)
- David N. Card, Published Sources of Benchmarking Data, memorandum, 5 pages, Software Productivity Consortium, March 2002.
- Deming, W. E. "Out of the Crisis". MIT Center for Advanced Engineering Study, Cambridge, MA. 1986.
- Donegan, P. M. "Medição de Qualidade de Software sob a Perspectiva do Six Sigma e CMMI". Monografia de conclusão do curso de Ciência da Computação da Universidade Estadual do Ceará, Fortaleza. 2005.
- Eckes, G. "Six Sigma for Everyone". John Wiley & Sons, Inc. 2001.
- Habib, M., Ahmed, S., Rehmat, A., Khan, M. J., & Shamail, S. (n.d.). Blending Six Sigma and CMMI - An Approach to Accelerate Process Improvement in SMEs. Framework, 386-391 p., 2008.
- Harry, M. e Schroeder, R. "Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations". Currency. 1998.
- Humphrey, W. S. "Managing the software process". Boston: Addison-Wesley. 1989.
- IEEE. "IEEE Standard 610.12-1990 Glossary of Software Engineering Terminology". IEEE CS Press. 1990.
- ISO 9001 (2000) "The International Organization for Standardization - Associação Brasileira de Normas Técnicas, NBR ISO 9001:2000 – Sistemas de gestão da qualidade – Requisitos", Rio de Janeiro, 2000.
- ISO 9004 (2000), "The International Organization for Standardization - Associação Brasileira de Normas Técnicas, NBR ISO 9004:2000 – Sistemas de gestão da qualidade – Diretrizes para melhorias de desempenho", Rio de Janeiro, 2000.
- ISO/IEC 12207 (1998) "Associação Brasileira de Normas Técnicas, NBR ISO/IEC 12207 – Tecnologia de Informação - Processos de ciclo de vida de software", Rio de Janeiro, 1998.
- ISO/IEC 15504-4 (2004) "The International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC 15504 - Information Technology - Process Assessment – Part 4: Guidance on using assessment results", 2004.
- ISO/IEC 15939 (2002) The International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC 15939 – Software engineering – Software measurement process, 37 pages.

- ISO/IEC TR 15504 (1998) “The International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC TR 15504 - Information Technology - Software Process Assessment”, document set with nine parts: ISO/IEC TR 15504-1 to ISO/IEC TR 15504-9, 1998.
- ISO/IEC TR 15504-7 (1998) “The International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC TR 15504-7 - Information Technology - Software Process Assessment - Part 7 : Guide for use in process improvement”, Technical Report, 1998.
- James Herbsleb, A. Carleton, J. Rozum, J. Siegel, David Zubrow, Benefits of CMM-Based Software Process Improvement: Initial Results CMU/SEI-94-TR-013, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA: 1994.
- McFeeley, Bob (1996) “IDEAL - A User's Guide for Software process Improvement”, Handbook CMU/SEI-96-HB-001, 236 pages, 1996.
- McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., Hall, F. (2002) Practical Software Measurement: Objective Information for Decision Makers, PSM, Addison Wesley Professional, ISBN: 0201715163, 304 p., 2002.
- Moreira, R. T. (2008) “Uma Abordagem para Melhoria do Processo de Software baseada em Medição”, dissertação de mestrado.
- Neil S. Potter and Mary E. Sakry (2002) “Making Process Improvement Work: A Concise Action Guide for Software Managers and Practitioners”, Addison-Wesley Professional, ISBN 0201775778, 2002.
- O’Toole, P. (2000) “Do’s and Don’ts of Software Process Improvements”, slides from tutorial presented at SIMPROS 2000 (Salviano e Santana 2000).
- Pande, P. S., Neuman, R. P. e Cavanagh, R. R. “The Six Sigma Way: How GE, Motorola, and Other Top Companies Are Honing Their Performance”. McGraw-Hill. 1998
- Paulk, Mark C., Weber, Charles V., Curtis, Bill and Chrissis, M. Beth (1994) “The Capability Maturity Model - Guidelines for Improving the Software Process”, CMU-SEI, Addison-Wesley, 441 pages, 1994.
- Pressman, R. “Software Engineering: A Practitioner’s Guide”. McGraw-Hill. 2002.
- Rasis, D., Gitlow, H.S., Popovich, E. Paper Organizers International: A Fictitious Six Sigma Green Belt Case Study I. Quality Engineering, 15 (1), pp.127-145, 2002.
- Rocha, Ana R. C. da, Maldonado, José C. e Weber, Kival C. (2001), “Qualidade de Software: Teoria e Prática”, Prentice Hall, 303 páginas, 2001.
- Salviano, C. F. “Uma Proposta Orientada a Perfis de Capacidade de Processo para Evolução da Melhoria de Processo de Software”. Tese (Doutorado) - Universidade Estadual de Campinas, Campinas, 2006.

Salviano, Clênio F. e Filho, Ozeas V. S. (Editores) (2000), Anais e Slides das Apresentações do SIMPROS 2000: Segundo Simpósio Internacional de Melhoria de Processo de Software, São Paulo, 442 páginas, 2000.

Scatolin, André Celso, Application of Six Sigma Methodology in order to Reduce Waste of a Manufacturing Process, Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2005. 137 p. Trabalho Final de Mestrado Profissional.

Sheard, Sarah A., “Evolution of the Frameworks Quagmire”, IEEE Computer, July 2001, pp. 96-98.

Shewhart, W. A. “Economic Control of Quality of Manufactured Product”. American Society for Quality Control. 1980.

Sommerville, I. “Software Engineering”. Addison-Wesley. 2006

Wang, H. (2008). A Review of Six Sigma Approach: Methodology, Implementation and Future Research. 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, 1-4. IEEE.

Yang, K. e El-Haik B. (2003) “Design for SIX SIGMA: A roadmap for product development”, McGraw-Hill. P. 12-13.

10.2	IDEAL.....	130
10.2.1	FASES DO IDEAL.....	132
10.2.1.1	FASE INICIAL (INITIATING).....	133
10.2.1.2	FASE DE DIAGNÓSTICO (DIAGNOSING).....	135
10.2.1.3	FASE DE ESTABILIZAÇÃO (DIAGNOSING).....	137
10.2.1.4	FASE DE AÇÃO (ACTING).....	139
10.2.1.5	FASE DE APROVEITAMENTO (LEVERAGING).....	141
10.2.1.6	FASE DE GERENCIAMENTO DO PROGRAMA DE MELHORIA DO PROCESSO DE SOFTWARE (MANAGE) 143	
10.3	PRO2PI.....	145
10.3.1	ENGENHARIA DE PROCESSO DIRIGIDA POR PERFIS DE CAPACIDADE E SEUS FUNDAMENTOS....	146
10.3.2	O PRO2PI.....	147
10.3.2.1	PRO2PI-PROP: PROPRIEDADES DE PRO2PI.....	152
10.3.2.2	PRO2PI-MODEL: MODELO DE PRO2PI.....	153
10.3.2.3	PRO2PI-MEAS: MEDIÇÕES PARA PRO2PI.....	155
10.3.2.4	PRO2PI-CYCLE: PROCESSO PARA CICLO DE MELHORIA.....	156
10.4	SEIS SIGMA.....	157
10.4.1	PDCA.....	160
10.4.2	DMAIC.....	160
10.4.2.1	DEFINIR.....	161
10.4.2.2	MEDIÇÃO.....	161
10.4.2.3	ANÁLISE.....	162
10.4.2.4	MELHORIA.....	162
10.4.2.5	CONTROLE.....	162
10.5	CONSIDERAÇÕES FINAIS.....	163
10.6	EXERCÍCIOS.....	164
10.7	SUGESTÕES DE LEITURA.....	164
10.8	TÓPICOS DE PESQUISA.....	165
	REFERÊNCIAS.....	166

Capítulo

10

Comment [j214]: Formatar todas a tabelas do capítulo

Qualidade de Produtos de Software

Este capítulo está dividido nas seguintes seções:

Seção 10.1 – Modelos de qualidade de produto: Serão apresentadas normas que objetivam garantir a qualidade do software.

Seção 10.2 – Teste de Software: Serão apresentados tipos, abordagens, estágios e processo de testes.

Seção 10.3 – Inspeção de Software: Nesta seção, serão apresentadas as etapas da inspeção de software e as ferramentas de apoio à inspeção.

Seção 10.4 – Modelos de Maturidade de Testes de Software: Serão abordados os modelos de maturidade de teste de software para avaliar e melhorar o nível de qualidade dos processos de testes aplicados numa organização desenvolvedora de software.

Introdução

Desde que a ABNT (Associação Brasileira de Normas Técnicas) foi criada, a preocupação com qualidade no Brasil vem se ampliando. A indústria busca continuamente aprimorar seus produtos e alinhar critérios com os padrões mais rigorosos em uso no mundo. A qualidade, atualmente, é percebida como um objetivo de negócio. Maior qualidade afinal significa cliente satisfeito.

Sob a perspectiva de software, o assunto qualidade é bastante extenso. Para cada aspecto diferente do ciclo de vida de um produto, há dezenas de técnicas e ferramentas visando apoiar os desenvolvedores. Existem soluções para agilizar tarefas, guiar profissionais ao aplicarem uma metodologia ou mesmo analisar o produto e suas especificações em busca de falhas potenciais.

Um problema fundamental da qualidade de software é definir claramente os objetivos que se pretende atingir com um projeto. Para fazê-lo, é preciso enumerar características desejáveis do software, que idealmente devem incluir valores quantitativos e qualitativos a serem respeitados.

Quando um produto é medido em relação a uma série de características, é possível realizar um diagnóstico mais preciso de sua qualidade. As medidas também podem ser usadas para uma definição mais precisa de requisitos, fixando-se no início do projeto, os valores desejados para o produto final.

10.1. Modelos de qualidade de produto

Os modelos de qualidade objetivam avaliar o produto de software, segundo diferentes aspectos baseados na visão do usuário. Para padronizar internacionalmente as características de implementação do software, foram criadas algumas normas que serão vistas a seguir.

10.1.1. ISO 9126

A norma 9126 é um conjunto de atributos que têm impacto na capacidade do software de manter o seu nível de desempenho dentro de condições estabelecidas por um dado período de tempo [Cortes 2009]. A ISO 9126 é dividida em quatro partes:

- ISO/IEC 9126-1:2001

Engenharia de Software – Qualidade do Produto - Parte 1: Modelo de Qualidade

- ISO/IEC TR 9126-2:2003

Engenharia de Software – Qualidade do Produto - Parte 2: Métricas Externas

- ISO/IEC TR 9126-3:2003

Engenharia de Software – Qualidade do Produto - Parte 3: Métricas Internas

- ISO/IEC TR 9126-4:2004

Engenharia de Software – Qualidade do Produto - Parte 4: Métricas de Qualidade em Uso

Este capítulo vai abordar as características da ISO/IEC 9126-1 pelo fato desta ser mais focada em modelo de qualidade, tema do presente livro. Serão apresentadas as diretrizes de uso e as características de qualidade de produto de software.

10.1.1.1. Diretrizes para uso da norma NBR ISO/IEC 9126-1

Esta norma pode ser aplicada nas etapas seguintes:

- Definição dos requisitos de qualidade de um produto de software;
- Avaliação da especificação de software para verificar se ele irá satisfazer aos requisitos de qualidade durante o desenvolvimento;
- Descrição de particularidades e atributos do software implementados, por exemplo, em manuais de usuário;
- Avaliação do software desenvolvido antes da entrega ao cliente;
- Avaliação do software desenvolvido antes da aceitação pelo cliente;

10.1.1.2. Características e subcaracterísticas de qualidade de software

Os modelos de qualidade, geralmente, representam a totalidade dos atributos do software classificados em uma estrutura de árvore hierárquica de características e subcaracterísticas. O nível mais alto dessa estrutura é composto pelas características de qualidade e o nível mais baixo é composto pelos atributos de qualidade do software.

Esta norma fornece um modelo de propósito geral, o qual define seis amplas categorias de características de qualidade de software, observadas na Tabela 10.1. que

O efeito combinado das características de qualidade em uma situação particular de uso é definido como qualidade em uso; também devem ser consideradas as qualidades internas e externas, detalhadas abaixo [Guerra & Colombo 2009].

- **Qualidade em Uso:** satisfazer as necessidades reais de usuários ao utilizar o produto de software, para atingir metas especificadas em contextos de uso especificados, ou seja, o efeito da utilização do produto, medido com relação às necessidades dos usuários.
- **Qualidade Externa:** influenciar o comportamento do sistema para satisfazer necessidades explícitas e implícitas, quando o sistema for utilizado sob condições especificadas, ou seja, o efeito da execução das funções medido com relação aos requisitos externos.
- **Qualidade Interna:** atributos estáticos do produto de software para satisfazer necessidades explícitas e implícitas, quando o sistema for utilizado em condições especificadas, ou seja, o efeito das propriedades de produtos intermediários, medidos com relação aos requisitos internos – projeto e código.

Tabela 10.1. Características de qualidade interna e externa (NBR ISSO/IEC 9126-1, 2001)

Características	Definições
Funcionalidade	Conjunto de atributos que evidencia a existência de um conjunto de funções e suas propriedades especificadas. As funções satisfazem às necessidades explícitas ou implícitas.
Confiabilidade	Conjunto de atributos que evidencia a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo definido.
Usabilidade	Conjunto de atributos que evidencia o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso por um conjunto de usuários.
Eficiência	Conjunto de atributos que evidencia o relacionamento entre nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas.
Manutenibilidade	Conjunto de atributos que evidencia o esforço necessário para fazer modificações necessárias no software.
Portabilidade	Conjunto de atributos que evidencia a capacidade do software de ser transferido de um ambiente para outro sem ter danos.

Comment [j215]: Deve ficar assim centralizado na célula e não como abaixo

Comment [j216]: Deve ficar assim Justificado na célula e não como abaixo

Além de definir as características de qualidade de software e orientar quando devem ser utilizadas, a norma também apresenta um modelo de qualidade de produto de software. Sua publicação teve o grande mérito de estabelecer um modelo básico de

Na Tabela 10.2 serão apresentadas as subcaracterísticas para cada característica e suas respectivas definições:

Tabela 10.2 - Características e subcaracterísticas de Qualidade

Características	Subcaracterísticas
Funcionalidade (satisfação das necessidades)	Adequação - execução do que é apropriado Acurácia - execução de forma correta Interoperabilidade - interação com outros sistemas Conformidade - aderência às normas Segurança de acesso - bloqueio de uso não autorizado
Confiabilidade (imunidade a falhas)	Maturidade - frequência das falhas Tolerância a falhas - reação a falhas Recuperabilidade - capacidade de se recuperar das falhas
Usabilidade (facilidade de uso)	Intelegibilidade - facilidade de entendimento Apreensibilidade - facilidade de aprendizado Operacionalidade - facilidade de operação
Eficiência (rápido e "enxuto")	Tempo - tempo de resposta, velocidade de execução Recursos - quais recursos são utilizados
Manutenibilidade (facilidade de manutenção)	Analisabilidade - facilidade de encontrar falha Modificabilidade - facilidade de modificar Estabilidade - baixo risco de alterações Testabilidade - facilidade de testar
Portabilidade (uso em outros ambientes)	Adaptabilidade - facilidade de se adaptar a outros ambientes Capacidade para ser instalado - facilidade de instalar em outros ambientes Conformidade - aderência a padrões de portabilidade Capacidade para substituir - facilidade de ser substituído por outro

Comment [j217]: Deve ficar assim centralizado na célula e não como abaixo

Comment [j218]: Deve ficar assim Justificado na célula e não como abaixo

10.1.2. ISO 12119

Esta norma foi publicada em 1994 e trata da avaliação de pacotes de software, também conhecidos como "software de prateleira". Além de estabelecer os requisitos de qualidade para este tipo de software, ela também destaca a necessidade de instruções para teste deste pacote. Esta norma divide-se em itens apresentados na Tabela 10.3.

Tabela 10.3. Itens da Norma 12119

1. Escopo
2. Definições
3. Requisitos de qualidade

3.1. Descrição do Produto	Descreve o produto, de forma a ajudar o comprador em potencial, servindo como base para testes. Cada declaração deve ser correta e testável. Deve incluir declarações sobre funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.
3.2. Documentação do usuário	Deve ser completa, correta, consistente, fácil de entender e capaz de dar uma visão geral do produto.
3.3. Programas e dados	Descreve em detalhes cada uma das funções do software, incluindo declarações sobre funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.
4. Instruções para teste	
4.1. Pré-requisitos de teste	Lista de itens necessários para o teste, incluindo documentos incluídos no pacote, componentes do sistema e material de treinamento.
4.2. Atividades de teste	Instruções detalhadas sobre os procedimentos de teste, inclusive instalação e execução de cada uma das funções descritas.
4.3. Registro de teste	Informações sobre como os testes foram realizados, de tal forma a permitir uma reprodução destes testes. Deve incluir parâmetros utilizados, resultados associados, falhas ocorridas e até a identidade do pessoal envolvido.
4.4. Relatório de teste	Relatório incluindo: identificação do produto, hardware e software utilizado, documentos utilizados, resultados dos testes, lista de não conformidade com os requisitos, lista de não conformidade com as recomendações, datas, etc.

Comment [j219]: Refazer essa tabela, pq alguns itens tem descrição e outros não, tabela muito confusa para leitor leigo

- Documentação do usuário de fácil compreensão;
- Presença de um Manual de instalação com instruções detalhadas;
- Possibilidade de verificar se uma instalação foi bem sucedida;
- Especificação de valores limites para todos os dados de entrada, que deverão ser testados;
- Operação normal mesmo quando os dados informados estão fora dos limites especificados;
- Função de auxílio (help) com recursos de hipertexto;
- Mensagens de erro com informações necessárias para a solução da situação de erro;
- Diferenciação dos tipos de mensagem: confirmação, consulta, advertência e erro;
- Clareza nos formatos das telas de entrada e relatórios;
- Capacidade de reverter funções de efeito drástico;
- Alertas claros para as conseqüências de uma determinada confirmação;
- Identificação dos arquivos utilizados pelo programa;
- Identificação da função do programa que está sendo executada no momento;
- Capacidade de interromper um processamento demorado;

10.1.3. ISO 14598

É um guia para avaliação de produtos de software, baseado na utilização prática da norma ISO 9126, já que esta define as métricas, características e subcaracterísticas de qualidade de software [Koscianski & Soares, 2007].

Esta norma possui recursos interessantes aos avaliadores, pois trata o processo de avaliação em detalhe. Ela leva em consideração a existência de três grupos interessados em avaliar um software, ou seja, os três tipos básicos de certificação, mostrados na Tabela 10.4:

Tabela 10.4. Tipos de certificação

Certificação	Quem realiza	Finalidade
de 1ª Parte	Empresas que desenvolvem software	Melhorar a qualidade de seu próprio produto
de 2ª Parte	Empresas que adquirem software	Determinar a qualidade do produto que irão adquirir
de 3ª Parte	Empresas que fazem certificação	Emitir documento oficial sobre a qualidade de um software

Esta norma se constituirá, na verdade, de seis documentos distintos, relacionados entre si, exibidos na Tabela 10.5:

Tabela 10.5. Documentos da ISO 14598

		Define os termos técnicos utilizados nas demais partes, contém requisitos gerais para especificação e avaliação de qualidade de software e esclarece os conceitos gerais.
14598-2	Planejamento e Gerenciamento	Sobre como fazer uma avaliação, de forma geral.
14598-3	Guia para Desenvolvedores	Como avaliar sob o ponto de vista de quem desenvolve.
14598-4	Guia para Aquisição	Como avaliar sob o ponto de vista de quem vai adquirir.
14598-5	Guia para Avaliação	Como avaliar sob o ponto de vista de quem certifica.
14598-6	Módulos de Avaliação	Detalhes sobre como avaliar cada característica.

As normas das séries 9126 e 14598 podem ser utilizadas em complementação, de acordo com o objetivo da avaliação. A norma NBR ISO/IEC 14598-1 contém conceitos de como avaliar a qualidade de software e define um modelo de processo de avaliação genérico. Juntas, as normas NBR ISO/IEC 14598-2 e NBR ISO/IEC 14598-6 estabelecem itens necessários para o suporte à avaliação. Por outro lado, as normas NBR ISO/IEC 14598-3, NBR ISO/IEC 14598-4 e NBR ISO/IEC 14598-5 estabelecem um processo de avaliação específico para desenvolvedores, adquirentes e avaliadores de software, respectivamente. O relacionamento entre elas pode ser visto na Figura 10.1.

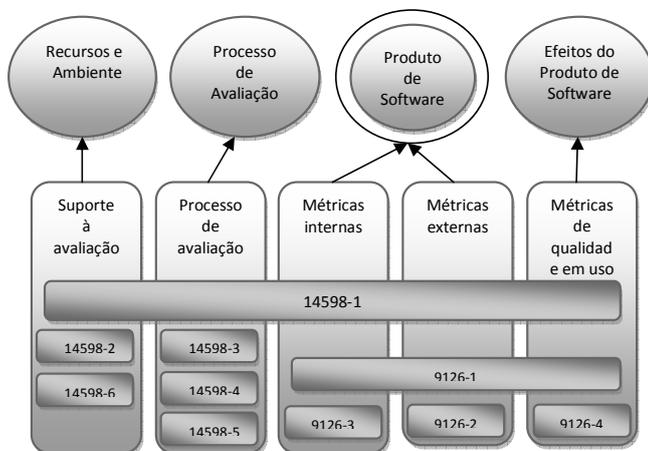


Figura 10.1. Relacionamento entre as séries 9126 e 14598

Em resumo, esta nova norma complementa a ISO/IEC 9126, permitindo que haja uma avaliação padronizada das características de qualidade de um software. É importante notar que, ao contrário da 9126, a 14598 possui detalhes mínimos, incluindo modelos para relatórios de avaliação, técnicas para medição das características, documentos necessários para avaliação e fases da avaliação. Como um exemplo, observe a Tabela 10.6 - um modelo de relatório de avaliação, segundo um

Seção	Itens
1 – Prefácio	Identificação do avaliador Identificação do relatório de avaliação Identificação do contratante e fornecedor
2 – Requisitos	Descrição geral do domínio de aplicação do produto Descrição geral dos objetivos do produto Lista dos requisitos de qualidade, incluindo: - Informações do produto a serem avaliadas - Referências às características de qualidade - Níveis de avaliação
3 - Especificação	Abrangência da avaliação Referência cruzada entre os requisitos de avaliação e os componentes do produto Especificação das medições e dos pontos de verificação Mapeamento entre a especificação das medições com os requisitos de avaliação
4 – Métodos	Métodos e componentes nos quais o método será aplicado
5 – Resultado	Resultados da avaliação propriamente ditos Resultados intermediários e decisões de interpretação Referência às ferramentas utilizadas

10.1.4. Projeto SQuaRE

O projeto SQuaRE (Software product Quality Requirements and Evaluation), traduzido: Requisitos de Qualidade e Avaliação de Produtos de Software surgiu a partir da necessidade de se criar um manual de utilização das normas ISO/IEC 9126 e ISO/IEC 14598. Foi feita uma reorganização do material existente sem mudanças radicais. O modelo hierárquico de qualidade proposto na 9126 continuou válido, assim como diversos aspectos organizacionais abordados na série 14598.

A norma SQuaRE surge de uma maneira muito sólida e por diversos motivos, pois abrange amplamente o assunto e estabelece uma base precisa, tanto para definir o modelo quanto para realizar a avaliação; Seus documentos contemplam um certo caráter didático: houve, por exemplo, a preocupação em fornecer uma extensa lista de exemplos de métricas; Os documentos são resultados de um esforço e consenso de centenas de pesquisadores; representam, assim, uma soma de experiências única no assunto; Outros modelos de qualidade elaborados por pesquisadores de maneira pontual podem ser mapeados para o modelo SQuaRE.

Comment [j220]: Falta referência

10.1.5. Norma SQuaRE

Na reorganização das antigas normas 9126 e 14598, o projeto SQuaRE adotou uma divisão de assuntos em cinco tópicos ilustrados na Figura 10.2.

Requisitos de Qualidade 2503n	Modelo de Qualidade 2501n	Avaliação 2504n
	Gerenciamento	



Figura 10.2. Partes componentes da norma SQuaRE

Cada divisão é composta por um conjunto de documentos e trata de um assunto específico, como apresentadas abaixo:

- **Gerenciamento:** os documentos desta divisão são voltados a todos os possíveis usuários dela, como gerentes, programadores, avaliadores ou compradores. São definidos os termos utilizados em todos os demais documentos e são feitas recomendações e sugestões de caráter geral sobre como utilizar o SQuaRE.
- **Modelo de Qualidade:** esta divisão corresponde principalmente à ISO/IEC 9126-1. São definidos os conceitos de qualidade externa, interna e em uso, que permitem orientar diferentes perspectivas de avaliação. Por exemplo, desenvolvedores e clientes têm visões e preocupações diferentes relacionadas à qualidade do mesmo produto. É definido um modelo hierárquico de características de qualidade, permitindo que se faça uma descrição extensa e precisa do que cada ator espera de um produto.
- **Medição:** dois pontos importantes fazem parte dessa divisão. Primeiro, definir o que é uma medição e descrever os diversos aspectos relacionados à realização dessa tarefa. Por exemplo: cuidados relacionados com a garantia da precisão dos resultados obtidos. O segundo ponto consiste na proposta de uma série de métricas que podem ser utilizadas ou adaptadas pelos usuários das normas às suas necessidades específicas.
- **Requisitos de Qualidade:** umas das noções importantes introduzidas pela 9126 e retomada pelo projeto SQuaRE é estabelecer objetivos de qualidade para um produto. Isto significa que para garantir a qualidade de um produto, apenas realizar medidas não basta: é preciso que metas tenham sido previamente especificadas. Tais valores fazem parte da especificação dos requisitos do software.
- **Avaliação:** a norma SQuaRE é concretizada na realização de uma avaliação de qualidade a partir de medições cujos resultados devem ser confrontados contra um modelo definido pelo usuário. A divisão de avaliação é direcionada aos diferentes públicos da norma, como desenvolvedores e compradores. São sugeridos procedimentos a serem adotados em cada caso para realizar uma avaliação.

Vale enfatizar que o projeto SQuaRE não surgiu para desmentir as normas 9126 e 14598, mas sim para organizá-las.

As características de qualidade das normas, citadas nesta seção, podem ser validadas e implementadas através de testes e inspeções de software que serão detalhados nas próximas seções.

Comment [j221]: Referencia?

Comment [j222]: Foi tirado de onde?

Comment [j223]: reescrever

Comment [j224]: usar outra palavra

10.2. Teste de Software

Quando um programa está sendo desenvolvido, atividades de checagem e análise devem ser realizadas durante todo o ciclo de vida do software. *Verificação e Validação* (V & V) é o processo utilizado para garantir que o produto atenda suas especificações e realizem a funcionalidade esperada pelo cliente. Verificação está relacionada à atividade de avaliar um produto e determinar se ele está de acordo com seus requisitos especificados, enquanto que validação possui um conceito de caráter mais abrangente: garantir que o produto atenda às necessidades dos clientes.

Dentro do processo de V & V existem duas abordagens para checar e analisar um sistema: Teste de Software e Inspeção de Software [Sommerville 2004]. A primeira é uma técnica dinâmica de verificação e validação, uma vez que seu foco está em exercitar e observar o comportamento operacional do produto. Já a Inspeção de Software verifica os artefatos de um sistema, como o documento de requisitos, diagramas de análise e projeto, e código fonte. Sendo assim, é uma técnica estática de V & V, já que não precisa que o software seja executado. Esta técnica será apresentada na próxima seção.

De um modo simples, um dos principais propósitos dos testes de software é o de executar um sistema desenvolvido de forma a encontrar defeitos que representem qualquer tipo de comportamento não desejado do sistema. Esta definição contrasta com o outro objetivo de testes: garantir que o sistema faz aquilo que é suposto fazer, ou seja, demonstrar ao cliente que o sistema atende aos seus requisitos funcionais e não funcionais.

O processo de teste de software é, portanto, importante para analisar se a implementação está em conformidade com os requisitos do sistema, para reduzir os custos associados à manutenção e retrabalho, verificar a integração correta entre todos os componentes do software, e especialmente, garantir o aumento da qualidade do software, resultando na satisfação do cliente.

A idéia de encontrar defeitos tem o intuito de reportá-los à equipe de desenvolvimento, de modo que eles possam ser corrigidos. Dessa forma, o produto final deve ter a menor quantidade de defeitos possível, garantindo assim sua qualidade e confiança. É importante ressaltar a impossibilidade de se encontrar todos os erros existentes em um programa através de testes. Sendo assim, é importante saber que os softwares sempre irão conter defeitos, e dessa forma, o objetivo é prover sistemas nos quais os defeitos remanescentes não sejam críticos a ponto de comprometer sua integridade.

De modo a deixar o processo de teste mais robusto e adequá-lo a boas práticas é interessante adotar um modelo de melhoria. Para isto, surgiram modelos de maturidade de teste para avaliar e melhorar os processos de teste de software nas organizações. Na seção 10.4 serão apresentados os 3 principais modelos. Nas próximas sub-seções serão apresentados conceitos fundamentais inerentes ao processo de teste de software.

10.2.1. Abordagens de Testes

Existem duas abordagens principais de testes: abordagem funcional (“black box” ou “caixa preta”) e abordagem estrutural (“White box” ou “caixa branca”) [Sommerville 2004], [Pressman 2002].

- **Caixa preta:** como o próprio nome já sugere, nesta abordagem o testador visualiza o software como uma caixa preta, ou seja, não considera a estrutura interna do programa, de que forma o código foi implementado ou que tecnologia foi utilizada, por exemplo. Considerando os dados de entrada, o objetivo principal é observar as saídas geradas pelo sistema e verificar se estas estão de acordo com o esperado. A Figura 10.3 ilustra este tipo de abordagem.

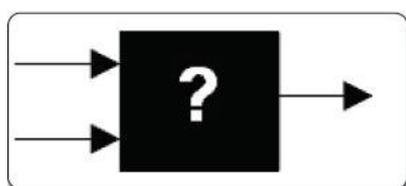


Figura 10.3 Abordagem funcional (Zeilinger 2003)

- **Caixa branca:** diferentemente da abordagem anterior, neste tipo de abordagem o testador está interessado no que está acontecendo “dentro da caixa”. É caracterizada por avaliar as funcionalidades internas dos componentes do software, baseando-se no código fonte e procurando exercitar estruturas de controle e de dados do programa. Sendo assim, faz-se necessário que o analista de testes tenha boa habilidade em programação de modo a entender todos os caminhos lógicos possíveis. A Figura 10.4 ilustra a abordagem estrutural.

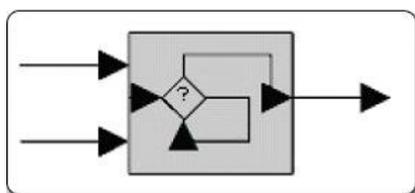


Figura 10.4 Abordagem estrutural (Zeilinger 2003)

10.2.2. Estágios de Testes

Os testes de software normalmente são executados em diferentes estágios durante o ciclo de vida do desenvolvimento do software. Dependendo do objetivo principal do teste, quatro estágios são conhecidos [Graham et. al 2007], [Myers 2004]:

- **Teste de unidade:** realiza testes em componentes individuais (módulos, programas, objetos, classes, etc) de forma a determinar se cada um deles, separadamente, está sendo executado de maneira correta. Normalmente estes testes são de caixa branca, realizados pelos próprios desenvolvedores do componente. Geralmente utilizam ferramentas que provêm um suporte adicional para checar a corretude do programa, como ferramenta de *debugging*

documentá-los formalmente, e assim, reduzindo o custo, pois antecipa a correção de defeitos. Geralmente é necessária a utilização de *stubs* (módulos que substituem outros módulos subordinados) e *drivers* (um módulo que substitui outro módulo que seja responsável por controlar a chamada de um sistema), para serem utilizados no lugar dos softwares que estejam eventualmente faltando e para simular a interface entre os componentes de software.

- **Teste de integração:** nesta etapa, as unidades que foram testadas individualmente no estágio anterior são testadas de forma integrada, bem como as interfaces entre os componentes. A integração deve ser realizada adicionando-se os componentes um por um, e após cada passo um teste é necessário (teste incremental). Esta técnica tem a vantagem de adiantar a detecção de defeitos no processo de testes e corrigi-los mais rapidamente, enquanto é mais fácil determinar as causas dos erros. Por outro lado, tem a desvantagem de ser uma prática bastante custosa. Sendo assim, a integração pode ser feita basicamente de duas formas: *Top-down* ou *Bottom-up*. Na primeira, os testes são realizados de cima para baixo (começando da GUI ou do menu principal); componentes ou sistemas são substituídos por *stubs*. Na segunda, os testes começam na parte mais básica do sistema até o nível mais alto; componentes ou sistemas são substituídos por *drivers*.
- **Teste de sistema:** neste nível o propósito do teste está em verificar o funcionamento de todo o sistema, já integrado, e analisar se ele está de acordo com os requisitos que foram especificados. Neste momento, não só são realizados os testes de integração dos componentes do software entre si, como também destes componentes com um ambiente de teste correspondente à produção final (hardware, software, outros sistemas), de modo a minimizar o risco de que falhas relacionadas com o ambiente operacional do produto não sejam encontradas. Geralmente a estratégia de caixa preta é utilizada neste estágio, mas testes de caixa branca também podem ser realizados.
- **Teste de aceitação:** o teste de aceitação corresponde ao teste realizado pelo usuário de fato do sistema, no momento em que todos ou quase todos os defeitos encontrados nas etapas anteriores já tenham sido corrigidos. O propósito deste teste é estabelecer a confiança do sistema; ele está mais relacionado com a validação do sistema, em que está se tentando determinar se o sistema está de acordo com os requisitos especificados. Normalmente os testes de aceitação podem ser de duas categorias: testes *alfa* e testes *beta*. Os primeiros são realizados nas instalações do desenvolvedor, que fica observando os usuários utilizarem o sistema, e anotam os problemas identificados. Já os testes *beta* são realizados no ambiente real de trabalho do usuário, que instala o sistema e testa, sem a presença do desenvolvedor. Em seguida, um documento contendo os registros dos problemas encontrados é enviado à organização desenvolvedora.

10.2.3. Tipos de Testes

Cada tipo de teste é focado em um grupo de atividades com um determinado objetivo. É necessário pensar em diferentes tipos de testes uma vez que testar a

num objetivo particular de teste, que poderia ser um teste de uma função a ser executada pelo componente ou sistema; alguma característica não funcional; a estrutura ou arquitetura do componente ou sistema, etc. Existem vários tipos de testes, dependendo do objetivo de cada projeto e de cada organização. Abaixo serão apresentados alguns dos mais comuns [Graham et. al 2007].

- **Teste funcional:** este tipo de teste está focado nas regras de negócio do sistema, ou seja, o fluxo de trabalho do programa é avaliado.
- **Teste de recuperação de falha:** o sistema é forçado a falhar de diversas maneiras de modo a verificar seu comportamento diante destas falhas, e reparar de que formas ele se recupera.
- **Teste de interoperabilidade:** testa um produto de software de modo a determinar sua capacidade de interagir com um ou mais componentes ou sistemas.
- **Teste de segurança:** verifica se o sistema possui atributos para prevenir acessos não autorizados, acidentais ou propositais, a programas e dados.
- **Teste de carga:** um tipo de teste para medir o comportamento do sistema quando este é submetido a níveis altos de carga, diferente das condições normais. É importante determinar o quanto de carga o sistema consegue suportar sem falhar.
- **Teste de performance:** verifica o rendimento de um sistema, como o tempo de resposta e processamento, taxa de transferência de dados, para diferentes condições (configurações, número de usuários, etc) as quais o programa é submetido.
- **Teste de estresse:** teste conduzido para avaliar o comportamento do sistema diante de condições que ultrapassem o limite especificado nos requisitos.
- **Teste de configuração:** testa o funcionamento do sistema em diferentes configurações de hardware/software, testando compatibilidade, configuração do servidor, tipos de conexões com a Internet, etc.
- **Teste de usabilidade:** testes para determinar se um produto é facilmente entendível, fácil de aprender, fácil de operar e atrativo aos usuários, ou seja, se o produto tem uma interface amigável para os que utilizarão o sistema.
- **Teste de regressão:** teste de regressão é a atividade de testar uma nova versão de um sistema para validar esta versão, detectando se erros foram introduzidos devido às mudanças realizadas no software, e então, garantir a correção das modificações. Uma vez que a re-execução de todos os testes é uma atividade bastante custosa, uma seleção de testes de regressão geralmente é realizada.

10.2.4. Processo de testes

O processo de testes pode ser dividido basicamente em cinco etapas: planejamento e controle, análise e projeto, implementação e execução, avaliação de critério de saída e reportagem e atividades de encerramento de testes [Graham et. al 2007]. Estas atividades são logicamente seqüenciais, porém, em um projeto específico, podem se sobrepôr, serem executadas em paralelo ou até mesmo serem

10.2.4.1. Planejamento e Controle

Durante o planejamento de testes deve-se ter certeza de que os objetivos dos clientes e *stakeholders* foram entendidos de maneira correta [Graham et. al 2007]. Baseados neste entendimento, os propósitos da atividade de testes propriamente dita são estabelecidos, e assim, uma abordagem e plano para os testes é obtida incluindo especificação das atividades de teste. O planejamento de testes apresenta as seguintes atividades principais:

- Determinar o escopo e riscos e identificar os objetivos de teste: são determinados os softwares, componentes, sistemas ou outros produtos que devem ser testados; os riscos que devem ser levados em consideração; e qual o propósito do teste (encontrar defeitos, verificar se está de acordo com os requisitos ou dentro dos padrões de qualidade, etc).
- Determinar a estratégia de teste: aqui serão estabelecidas as técnicas que serão utilizadas, o que precisa de fato ser testado (selecionar e priorizar os requisitos) e que nível de cobertura é necessário. Serão também analisadas quais pessoas precisarão se envolver e em que momento (desenvolvedores, usuários, etc), incluindo a definição da equipe de teste.
- Definir recursos: são definidos todos os recursos necessários durante o ciclo de vida de testes, tanto recursos materiais (PCs, software, ferramentas, etc) como recursos humanos (principais e de apoio).
- Fazer um cronograma para análise e projeto, implementação, execução e avaliação de teste: deverá ser elaborado um cronograma de todas as tarefas e atividades, para que seja possível terminar a fase de testes a tempo.
- Estabelecer os critérios de saída: critérios de saída, como critério de cobertura, por exemplo, deverão ser estabelecidos de modo a determinar quando a etapa de testes chegou ao fim.

Após planejar é necessária uma medida de controle para verificar se tudo está indo de acordo com o planejado. É preciso comparar o andamento real com o que foi estabelecido no plano de testes, e tomar medidas corretivas quando necessário.

10.2.4.2. Análise e Projeto

Esta é a atividade em que os objetivos gerais de testes são transformados em condições e projetos de teste tangíveis [Graham et. al 2007]. O propósito principal é identificar e descrever os casos de teste para cada versão de teste, e identificar e estruturar os procedimentos de teste, especificando como executar os casos de teste. As principais tarefas desta etapa podem ser destacadas em:

- Revisar a base de testes (como a análise de risco do produto, requisitos, arquitetura, especificação de projeto, e interfaces): a base de testes é utilizada para criar os testes. É possível começar a projetar os testes de caixa preta antes da implementação, uma vez que a base de testes pode ser usada para compreender o que o sistema precisa fazer.
- Identificar e descrever casos de teste: um caso de teste é um cenário associado a um requisito; é um texto contendo: identificador, objetivo, pré-condições de

Comment [j225]: referência muito repetida, buscar outras fontes, para esta e para as demais seções

Comment [j226]: referência muito repetida, buscar outras fontes, para esta e para as demais seções

- Estruturar procedimentos de teste: o passo a passo que descreve como os casos de teste devem ser executados. Inclui o estado inicial da aplicação; condições de funcionamento; como e quando fornecer os dados de entrada e obter os resultados; a forma de avaliar estes resultados, dentre outros.
- Avaliar a capacidade de testar os requisitos: a especificação de requisitos deve ser completamente clara, informando as condições necessárias para se definir os testes. Por exemplo, se a performance do software é algo crítico, deve ser claramente especificado o tempo de resposta mínimo em que o sistema deve responder.

10.2.4.3. Implementação e Execução

Uma vez que os casos e procedimentos de teste foram especificados em alto nível na etapa anterior, este é o momento em que o ambiente será preparado para que eles sejam executados e comparados com os resultados desejados [Graham et. al 2007]. Além disso, é a etapa em que os componentes necessários são implementados para que os testes sejam executados. As principais tarefas destas duas fases serão destacadas a seguir.

- Implementação:
 - Implementar componentes: efetuar a implementação de novos componentes de apoio necessários à aplicação dos testes, ou modificação de componentes já existentes. Ferramentas de automação podem ser utilizadas ou os componentes podem ser desenvolvidos explicitamente.
 - Criar suítes de teste: baseado nos casos de teste, um conjunto de testes que naturalmente trabalham juntos, forma uma suíte de teste e são utilizados para uma execução de teste eficiente.
 - Implementar e verificar o ambiente: preparar e verificar se o ambiente de teste está funcionando corretamente.
- Execução:
 - Executar as suítes de teste e casos de teste individuais, de acordo com os procedimentos de teste. Pode ser feito manualmente ou com o auxílio de ferramentas de execução de testes.
 - Seguir as estratégias de teste definidas na etapa de planejamento.
 - Criar um *log* com as saídas da execução dos testes e registrar os identificadores e versões do software que está sendo testado.
 - Fazer a comparação dos resultados esperados e dos resultados obtidos.
 - Quando houver diferenças entre os resultados esperados e os resultados obtidos, registrar os defeitos em um repositório centralizado. Não se deve registrá-los de forma aleatória.
 - Realização de testes de regressão para confirmar que uma falha anteriormente registrada foi de fato consertada.

Comment [j227]: referência muito repetida, buscar outras fontes, para esta e para as demais seções

Esta é a fase em que se deseja observar se já foram executados testes suficientes para garantir a qualidade desejada do produto, sendo assim, critérios de saída são definidos com esta finalidade [Graham et. al 2007]. Estes critérios informam se uma dada atividade de testes pode ser considerada completa. As principais atividades são:

- Checar se os *logs* de testes batem com os critérios de saída especificados no plano de testes: procura-se pelos testes que tenham sido executados e avaliados, e se defeitos foram encontrados, consertados ou re-testados.
- Verificar se será necessária a inclusão de mais testes ou se os critérios de saída especificados devem ser mudados: mais casos de testes podem precisar ser executados, se por acaso estes não tiverem sido todos executados conforme esperado, ou se for detectado que a cobertura de requisitos necessária ainda não foi atingida, ou até mesmo se aumentaram os riscos do projeto.
- Escrever um relatório de resumo de testes para os *stakeholders*: todos os *stakeholders* devem saber quais testes foram executados e quais os resultados destes testes, de modo a perceber que decisões precisam ainda ser tomadas visando a melhoria da qualidade do software.

10.2.4.5. Atividades de encerramento de teste

A atividade de encerramento de teste pode ser dada através de diversos fatores, como por exemplo, as informações necessárias do processo de testes já foram atingidas; o projeto é cancelado; quando um marco particular é alcançado; ou quando uma versão de manutenção ou atualização está concluída [Graham et. al 2007]. As atividades principais são:

- Checar se as entregas que foram programadas foram de fato entregues e garantir que todos os problemas reportados foram realmente resolvidos. Para os que permaneceram em aberto devem-se requisitar mudanças em uma futura versão.
- Finalizar e arquivar os artefatos produzidos durante o processo necessário para planejar, projetar e executar testes, como por exemplo, documentação, scripts, entradas, resultados esperados, etc. É importante reutilizar tudo que for possível destes artefatos, pois assim se consegue economizar tempo e esforço do projeto.
- Repassar os artefatos anteriormente citados para a equipe de manutenção, que irá prover suporte aos usuários do sistema e resolver qualquer problema encontrado depois de sua entrega.
- Avaliar como se deu o processo de testes e analisar as lições aprendidas, que serão de grande utilidade para futuras versões dos projetos. Este passo pode permitir não só melhorias no processo de testes, como também melhorias no processo de desenvolvimento do software como um todo.

10.2.5. Testes ao longo do ciclo de vida de Software

As atividades de teste não são atividades que são realizadas sozinhas, mas sim em paralelo com o ciclo de vida de desenvolvimento do software. Dessa forma, a

Comment [j228]: referência muito repetida, buscar outras fontes, para esta e para as demais seções

Comment [j229]: referência muito repetida, buscar outras fontes, para esta e para as demais seções

projeto. Portanto, o modo como as atividades de teste são estruturadas deve se ajustar ao modelo de desenvolvimento de software, ou do contrario, não conseguirá obter o sucesso desejado.

Um modelo de desenvolvimento de software bastante conhecido é o modelo em cascata, que como o próprio nome já sugere, tem sua base voltada a um desenvolvimento seqüencial das atividades. As primeiras atividades começam no topo da cascata, e então vão seguindo seqüencialmente através das várias atividades de concepção do projeto, e finalmente terminando com a etapa de implementação. Após isso, é que as atividades de teste são introduzidas, e dessa forma os defeitos só podem ser detectados bem perto da fase de implementação [Graham et. al 2007]. A Figura 10.5 ilustra o modelo em cascata.

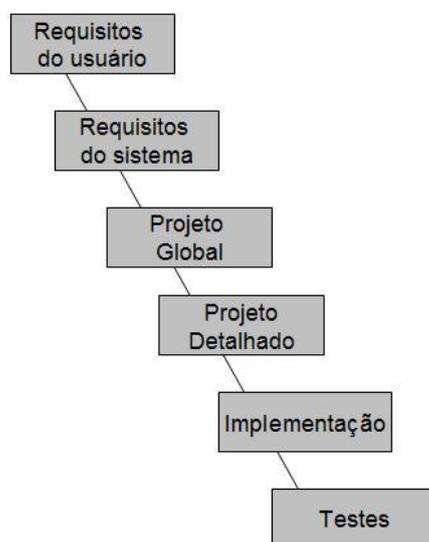


Figura 10.5 Modelo em cascata (adaptado de Graham 2007)

Com o objetivo de tentar contornar os problemas do modelo em cascata, foi desenvolvido o modelo V, que foca nos testes do produto durante todo o ciclo de desenvolvimento para conseguir uma detecção adiantada de defeitos. A idéia é que as atividades de testes não são simplesmente uma fase única, mas pelo contrário, como já foi visto na sessão anterior, se faz necessária toda uma preparação, passando por etapas de planejamento, análise, projeto, etc, que devem ser executadas em paralelo com as atividades de desenvolvimento.

Todos os artefatos gerados pelos desenvolvedores e analistas de negócio durante o desenvolvimento, provêm a base de testes em um ou mais níveis. Promovendo as atividades de teste mais cedo, defeitos podem ser geralmente encontrados nos documentos da base de testes. O modelo V demonstra como as atividades de verificação e validação podem ser executadas em conjunto com cada fase do ciclo de vida de desenvolvimento do software.

Basicamente, o ciclo de desenvolvimento do modelo usa os quatro estágios de teste

Comment [j230]: referência muito repetida, buscar outras fonts, para esta e para as demais seções

Comment [j231]: et al?

acordo com seus objetivos específicos. Na prática, os estágios de testes podem variar no modelo V dependendo dos propósitos do projeto, podendo ser combinados ou reorganizados. O modelo V pode ser visualizado na Figura 10.6.

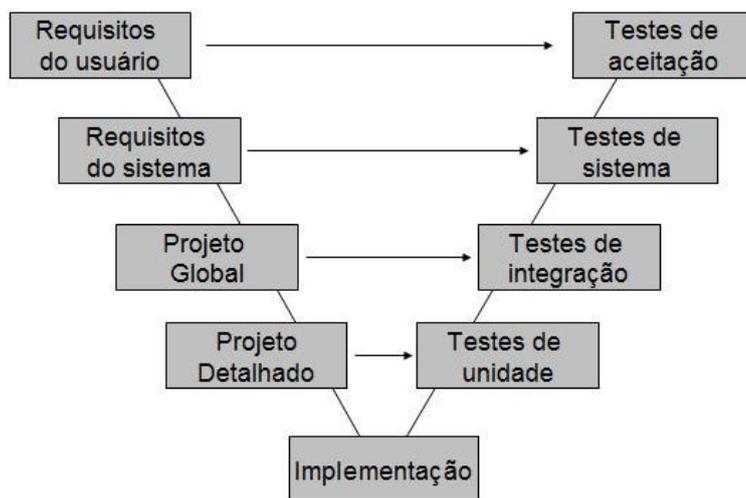


Figura 10.6 Modelo V (adaptado de Graham 2007)

Comment [j232]: Et al?

10.3. Inspeção de Software

Como explicado na sessão anterior, a inspeção de software é uma técnica estática do processo de V & V, em que são efetuadas revisões no sistema com o objetivo de encontrar defeitos e então, corrigi-los. O objetivo principal das inspeções é garantir que defeitos sejam reparados o mais cedo possível no processo de desenvolvimento de software, uma vez que quanto mais evoluído o software estiver, mais difícil será para encontrar os erros e mais custoso ainda consertá-los. Qualquer artefato produzido no desenvolvimento do software pode ser utilizado no processo de inspeção, como requisitos, modelo de projeto ou código.

O modelo CMMI exige a realização de revisões como uma prática específica do processo de verificação, demonstrando assim sua importância na garantia da qualidade do produto. Segundo Fagan, a utilização de inspeções informais de software captura em torno de 60% dos erros em um programa [Fagan 1986]. Mills et al. sugere que uma aplicação mais formal de inspeção de software pode detectar até mais de 90% dos erros de um programa [Mills et al. 1987]. Selby e Basili comparam empiricamente a efetividade de inspeções e testes. Eles perceberam que a revisão de código estática se mostrava mais efetiva e menos cara do que a procura por erros utilizando testes [Selby et al. 1987].

Comment [j233]: Substituir por et. al

Boehm e Basili ressaltam ainda que inspeções e testes capturam diferentes tipos de defeito e em diferentes momentos do processo de desenvolvimento de software [Boehm e Basili 2001]. Dessa forma, é interessante aplicar tanto inspeções como testes para detectar defeitos em artefatos de software. Começando o processo de V & V com inspeções, defeitos podem ser encontrados e corrigidos nas etapas iniciais do processo de desenvolvimento do software, e uma vez que o sistema esteja

10.3.1. A Equipe de Inspeção (Participantes)

A equipe de inspeção é composta por um pequeno grupo de pessoas que possuam interesse e conhecimento do produto. Geralmente o tamanho da equipe varia de quatro a sete participantes, e o número mínimo é de três pessoas. Equipes maiores são normalmente utilizadas para analisar documentos de mais alto nível do produto, enquanto que times menores são preferíveis ao se inspecionar detalhes mais técnicos.

É bastante interessante para o processo de inspeção que exista uma boa variedade de inspetores, pertencentes a diferentes áreas de conhecimento. O papel de cada participante será explicado abaixo.

- **Autor:** é o criador (desenvolvedor) do artefato que será inspecionado. Suas principais responsabilidades são: corrigir os problemas detectados durante o processo de inspeção, prover uma visão geral do produto aos demais participantes e tirar quaisquer dúvidas que surgirem com relação ao artefato desenvolvido.
- **Inspetor:** examina o produto antes e durante a reunião de inspeção (fase de preparação) de modo a tentar encontrar defeitos. Pode também identificar problemas amplos que estão fora do escopo da equipe de inspeção, como também sugerir melhorias.
- **Leitor:** pessoa responsável por apresentar o artefato aos demais participantes do processo de inspeção durante a reunião. Uma pessoa que usará o produto numa próxima etapa do seu ciclo de vida é um candidato forte para esta tarefa, uma vez que a atividade de ler sobre o produto irá permitir a este potencial usuário se tornar bastante familiar com o produto.
- **Escritor:** tem o papel de registrar as informações sobre cada defeito encontrado durante a reunião, que incluem: a localização do defeito, um resumo do problema, sua classificação e uma identificação do inspetor que o encontrou. Todas as decisões e recomendações feitas também são registradas.
- **Moderador:** o moderador tem o papel mais crítico no processo de inspeção e por este motivo faz-se necessário um treinamento mais aprofundado do que os outros membros da equipe. Ele é a pessoa que lidera toda a equipe e participa ativamente de todas as etapas. Dentre suas principais responsabilidades podemos destacar: selecionar e liderar a equipe de inspeção, distribuir o material a ser inspecionado, agendar as reuniões, atuar como moderador nos encontros, supervisionar a correção dos defeitos, e emitir relatório de inspeção. Uma outra responsabilidade muito importante do moderador é garantir que o foco da reunião se mantenha em encontrar falhas no produto, e não em acusar o autor dos problemas encontrados.

10.3.2. O Processo de Inspeção de Software (Etapas)

O processo tradicional de inspeção de software [Fagan 1976] é definido por seis estágios, cada um representado por seu principal responsável. A Figura 10.7 ilustra esta sequência de etapas e em seguida cada uma das etapas será explicada

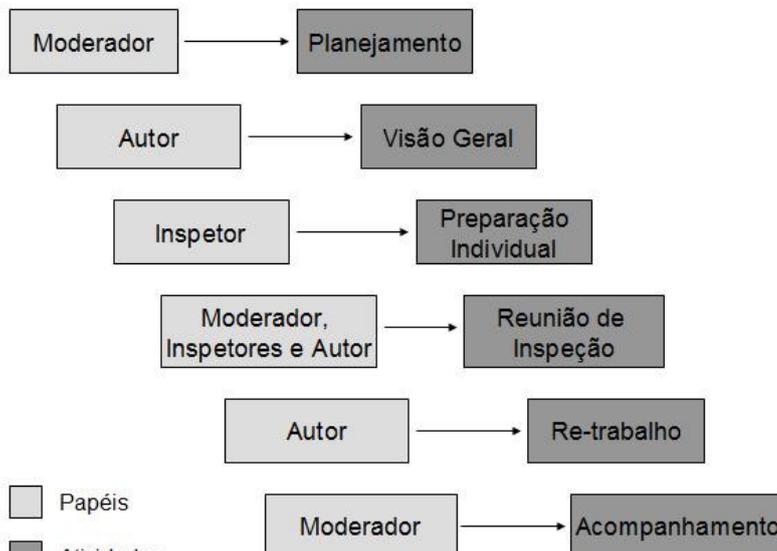


Figura 10.7 Processo de Inspeção de Software (adaptado de Fagan, 1976)

- Planejamento:** o moderador é a pessoa responsável por esta etapa. O planejamento envolve selecionar a equipe, checar se o produto está pronto para inspeção, organizar a reunião, delegar as atividades de cada membro e garantir a completude dos materiais a serem inspecionados. Nesta etapa o moderador também deve verificar se o material a ser inspecionado possui um tamanho adequado para uma única reunião. Caso contrário, o material deverá ser dividido em tamanhos menores, com inspeções a serem realizadas para cada uma destas partes.
- Visão geral:** nesta etapa o autor apresenta o produto aos demais membros da equipe, descrevendo o que o programa é suposto fazer. O moderador é responsável por decidir se esta etapa se faz realmente necessária, pois se a equipe já for bem familiarizada com o material a ser inspecionado ou novas técnicas não estejam sendo aplicadas, este estágio é dispensável.
- Preparação:** este é o momento em que cada membro do time de inspeção estuda individualmente a especificação e o programa a ser inspecionado, e procura por defeitos no material. Todos os possíveis defeitos devem ser registrados num *log* de preparação, assim como o tempo que foi gasto na preparação. O moderador é encarregado de analisar os *logs* antes da reunião de inspeção para determinar se a equipe está preparada para suas tarefas, e caso contrário, ele pode remarcar a reunião.
- Reunião:** nesta etapa, o passo a passo principal consiste na leitura e interpretação do produto, pelo leitor; em seguida o autor tira quaisquer dúvidas que eventualmente surgirem com relação ao material, e a equipe de inspetores então identificam os possíveis defeitos. Esta reunião deve ser curta, não podendo passar mais do que duas horas, e deve ser focada na detecção de

de inspeção não deve discutir como estes defeitos poderiam ser corrigidos e nem sugerir mudanças em outros componentes.

- **Re-trabalho:** o propósito do re-trabalho é corrigir os defeitos identificados durante a reunião de inspeção. O autor é a pessoa responsável por essas correções, devendo corrigir em primeiro lugar os defeitos considerados mais relevantes e graves, e corrigindo os de menor importância apenas se o tempo permitir.
- **Acompanhamento:** aqui o moderador deve decidir se uma nova inspeção é necessária ou não. Ele deve analisar o material corrigido pelos autores e verificar se os defeitos foram corrigidos com sucesso. O moderador pode incluir revisores adicionais nesta etapa se forem necessários conhecimentos técnicos extras. Se todos os problemas mais relevantes forem resolvidos, os problemas em aberto solucionados, e o produto satisfizer aos critérios de saída, o moderador aprova o *release* do produto. Se as condições não foram atingidas, ainda será necessário mais tempo na etapa de re-trabalho.

10.3.3. Ferramentas de Apoio ao Processo de Inspeção

Baseado na classificação de *groupware* (softwares voltados para o apoio a atividades de trabalho em grupo) e nas constantes mudanças tecnológicas, [Hedberg 2004] identificou quatro gerações de ferramentas de inspeção de software:

1. Primeiras Ferramentas (*Early tools*)
2. Ferramentas Distribuídas (*Distributed tools*)
3. Ferramentas Assíncronas (*Asynchronous tools*)
4. Ferramentas baseadas em WEB (*Web-based tools*)

Nota-se que as primeiras ferramentas a surgirem foram classificadas como Primeiras Ferramentas, no início da década de 90 e logo em seguida vieram as Ferramentas Distribuídas. No final da década de 90 surgiram as ferramentas para Internet.

As ferramentas da primeira geração são aquelas que apenas permitem o trabalho de toda a equipe no mesmo ambiente e ao mesmo tempo (inspeções síncronas). A segunda já permite que a equipe possa trabalhar de forma distribuída, ou seja, em lugares diferentes, porém ainda é preciso que seja ao mesmo tempo (inspeções distribuídas). A total independência de tempo e lugar foi introduzida na terceira geração, com as ferramentas assíncronas. As ferramentas da quarta geração também são assíncronas, diferenciando-se das demais devido a sua base tecnológica.

A seguir será apresentada uma ferramenta representante de cada geração introduzida anteriormente [Wong 2006]. A ferramenta ICICLE representará a geração de Primeiras Ferramentas. Em seguida a ferramenta Scrutiny exemplificará as Ferramentas Distribuídas. Assist ilustrará as Ferramentas Assíncronas, e finalmente, IBIS será a representante das Ferramentas baseadas em WEB.

- ICICLE – O ICICLE (*Intelligent Code Inspection Environment in a C Language Environment*) é o primeiro software de revisão publicado e visa apoiar o processo tradicional de inspeção de software. Como o próprio nome

tanto nas fases de preparação individual como nas reuniões em grupo. A reunião de inspeção em grupo deve ser realizada no mesmo local e a inspeção individual permite entrar com comentários em cada linha de código. A ferramenta não se aplica a inspeções mais genéricas, limitando o tipo de artefato a ser inspecionado e a técnica de detecção de defeitos, mas pode, entretanto, ser utilizada para inspecionar linhas de texto numa análise inicial. Um dos principais objetivos desta ferramenta é o de ajudar os inspetores de código a encontrarem defeitos óbvios.

- **Scrutiny** – O Scrutiny é uma ferramenta colaborativa *online*, sendo a primeira a permitir que os membros do time de inspeção se encontrassem dispersos geograficamente, podendo ser usada tanto de forma síncrona como assíncrona. Ela pode ser integrada com outras ferramentas e customizada para apoiar diferentes processos de desenvolvimento. Atualmente apenas suporta inspeções de textos. A ferramenta é baseada num processo de inspeção dividido em quatro etapas. No primeiro estágio, de iniciação, o moderador disponibiliza o documento a ser inspecionado na ferramenta. No próximo estágio, preparação, os inspetores inserem seus comentários a serem discutidos na reunião. Depois, na fase de resolução, o moderador guia os inspetores através dos documentos e dos defeitos coletados. Finalmente, no estágio de finalização, após as discussões e acordos referentes aos defeitos levantados, a ferramenta fornece um resumo dos defeitos que foram discutidos.
- **Assist** – *Asynchronous/ Synchronous Software Inspection Support Tool* foi desenvolvida para prover inspeções individuais e em grupo. Como o nome sugere, permite inspeções síncronas e assíncronas, com reuniões tanto em locais diferentes como no mesmo ambiente. Utiliza uma linguagem de definição de processo de inspeção (IPDL) e um sistema flexível para o tipo de documento inspecionado, permitindo o suporte a qualquer tipo de processo de inspeção de software. Inspeção de código, coletas de dados para métricas e cálculos para apoio as inspeções também estão presentes nesta ferramenta. É baseada numa arquitetura cliente/servidor, em que o servidor é usado como um repositório central de documentos e outros tipos de dados. Um *browser C++* pode automaticamente apresentar itens relevantes de *checklist*¹³ para a sessão de código inspecionado.
- **IBIS** – *Internet-Based Inspection System* é uma ferramenta baseada em WEB com notificações por *email* que auxilia no processo de inspeção desenvolvido por Fagan. Permite que as inspeções sejam realizadas entre pessoas geograficamente distribuídas e possui uma interface bastante leve e amigável, tendo toda sua estrutura e dados armazenados em arquivos XML. Ela não limita o tipo de artefato a ser inspecionado e provê suporte a decisões, apoio a anotações e *checklists*. As principais vantagens desta ferramenta são: permite que os inspetores acessem a aplicação de seus próprios computadores; admite que a inspeção seja realizada com integrantes da equipe distribuídos em locais diferentes, até mesmo em países diferentes; permite que especialistas diferentes participem da reunião, podendo ser especialistas de outro departamento ou mesmo fora na organização.

10.4. Modelos de Maturidade de Testes de Software

Para se construir software com qualidade, é necessário que se tenha um processo de testes bem definido e que ele esteja alinhado ao processo de desenvolvimento. Nesta seção serão vistos três modelos de maturidade de teste de software, os quais indicam como criar e/ou melhorar o processo de testes.

10.4.1. Processo de melhoria de testes – TPI

O modelo TPI foca na melhoria do processo de testes e ajuda a definir gradualmente os passos para sua evolução, levando em consideração o tempo, o custo e a qualidade. Foi desenvolvido porque a IQUIP (Intelligent & Quick Information Processing) e seus clientes necessitavam de um modelo que suportasse as constantes mudanças do processo de testes.

O processo de teste é parte do processo total de desenvolvimento, demonstrado na Figura 10.8, por isso é de extrema importância analisar os problemas relacionados às atividades de teste e os impactos que eles causam no processo geral [JACOBS & Sotokes 2000].

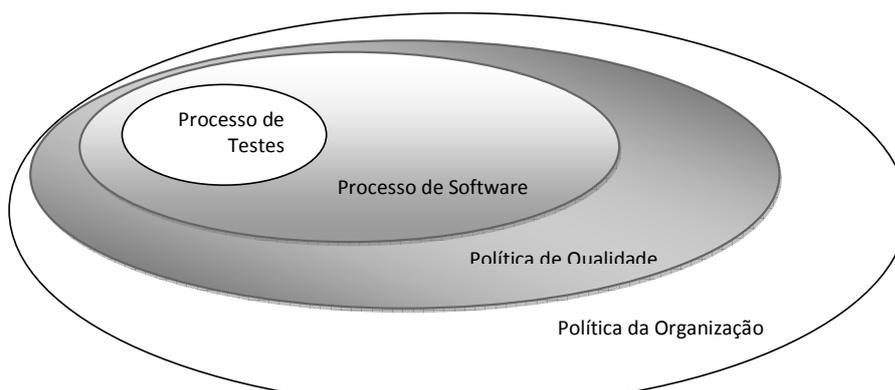


Figura 10.8. Processo Total de Desenvolvimento

Comment [j234]: referencia

10.4.1.1. Escopo do TPI

O TPI possui vinte áreas chave dentro do processo de testes, em que cada área tem níveis que representam a maturidade do processo de testes utilizado na empresa, checkpoints que são os pontos verificados para definir onde o processo se encaixa e em que ponto a empresa quer chegar e as sugestões de melhoria, em que se explicam quais são os passos a seguir para se alcançar o nível desejado. A estrutura do TPI pode ser observada na Figura 10.9.

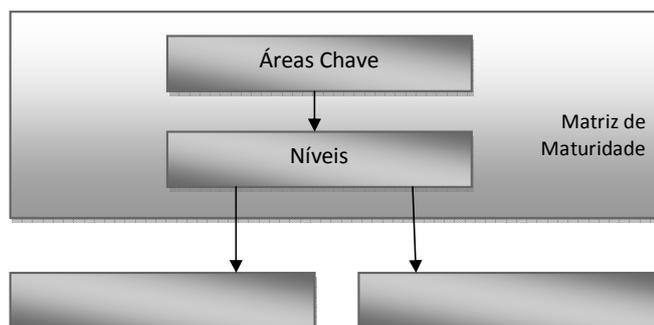


Figura 10.9. Estrutura do TPI

10.4.1.2. Áreas Chave

O TPI possui vinte áreas chave, descritas abaixo, divididas em ciclo de vida do desenvolvimento do software, técnicas de planejamento e testes, infraestrutura do ambiente de testes e fatores organizacionais.

- **Estratégia de teste:** É focada na detecção de defeitos importantes o mais cedo possível. A estratégia define que requisitos e riscos serão cobertos por quais testes.
- **Modelo de ciclo de vida:** Definição das fases, como planejamento, preparação, especificação, execução e finalização. Em cada fase, várias atividades são requeridas.
- **Quando se envolver?** O envolvimento tardio dos testes é um risco para o projeto, porque os defeitos são mais caros e mais difíceis de serem consertados. Por esta razão, o TPI defende que os engenheiros de teste devem se envolver no início do projeto.
- **Estimativa e planejamento:** Indica que atividades terão que ser feitas e que recursos serão necessários para um certo período de tempo.
- **Técnicas de especificação de testes:** Padronização do caminho para especificar casos de testes da fonte de informação. Aplicando estas técnicas, obtém-se uma melhor qualidade e reusabilidade dos casos de testes.
- **Técnicas de teste estático:** Nem tudo pode ser testado dinamicamente, ou seja, rodando programas. Inspeção de produtos, avaliação de medidas ou checklists são exemplos de técnica de teste estático.
- **Métricas:** Para o processo de testes, as métricas de progresso e de qualidade do sistema testado são muito importantes. Especificamente, para melhoria do processo de testes, as métricas devem avaliar as conseqüências de certas ações de melhoria.
- **Automação de testes:** Quando possível, a automação serve para diminuir as horas de execução e ter um status dos resultados, o quanto antes.
- **Ambiente de Testes:** Hardware, software, conexão ao banco de dados etc. O ambiente de testes tem uma grande influência na qualidade do software.
- **Ambiente de trabalho:** Um desorganizado ambiente de trabalho pode desmotivar as pessoas e, conseqüentemente, impactar o resultado dos testes.
- **Comprometimento e motivação:** A área de testes deve ter suficiente tempo, dinheiro e recursos (quantidade e qualidade) para que os profissionais realizem bons testes.
- **Funções de testes e treinamento:** Para a área de testes, uma mistura de diferentes conhecimentos, funções e perfis é requerida. É necessário ter experiência em testes. boa comunicação. conhecimento sobre o projeto e

- **Escopo da metodologia:** A empresa deve usar uma metodologia suficientemente genérica para ser aplicada em diversas situações e não tenha a necessidade de ser remodelada a cada novo projeto.
- **Comunicação:** No processo de teste, a comunicação é feita em diversos níveis, dependendo das pessoas envolvidas (desenvolvedor, usuário, cliente, gerente, etc).
- **Reportagem:** Teste não é só detecção de defeitos, mas sim a principal maneira de melhorar a qualidade de software. O relatório de resultados deve, não só conter as causas dos defeitos, como também o melhor caminho para consertá-lo.
- **Gerenciamento dos defeitos:** Deve ser capaz de acompanhar o ciclo de vida do defeito, além de dar suporte à análise e a resolução dos mesmos.
- **Gerenciamento dos documentos de testes:** Os produtos (plano de testes, especificações, banco de dados e arquivos) de teste devem ser mantidos, reusáveis e gerenciáveis.
- **Gerenciamento do processo de testes:** Para gerenciar as atividades de teste, é essencial seguir esses quatro passos: planejar, fazer, checar e atuar. O gerenciamento é vital para a realização dos testes.
- **Avaliação:** Começa na inspeção de produtos intermediários de acordo com requisitos. Os defeitos são encontrados em estágios iniciais e o re-trabalho tem um custo baixo.
- **Testes de baixo nível:** Testes unitários e de integração. Erros encontrados na fase de desenvolvimento.

10.4.1.3. Passos para implantar a melhoria

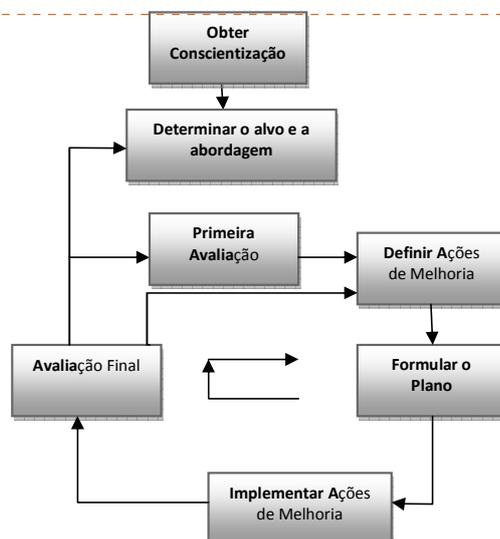


Figura 10.10. Modelo para implantar Melhorias

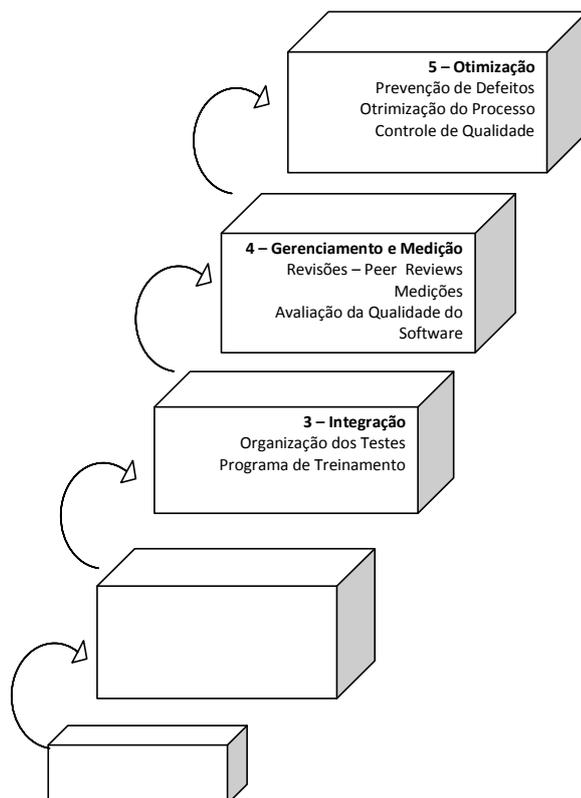
Como pode ser observado na Figura 10.10, há uma série de passos, detalhados abaixo, para se implantar a melhoria do processo de testes. É preciso fazer uma análise de

Comment [j235]: corrigir setas soltas no meio da figura e colocar referencia

- Obter conscientização: saber que o processo precisa melhorar e ter compromisso, não só no início das mudanças, mas sim ao longo de todo o projeto.
- Determinar o alvo e a abordagem: que áreas serão atacadas e quais as metas de melhoria?
- Primeira avaliação: conhecimento da situação atual. A partir da matriz são verificados os pontos fortes e fracos do processo de teste. Com base em entrevistas e documentação, os níveis das áreas-chave do TPI são definidos.
- Definir ações de melhoria: baseadas nas metas e no resultado da avaliação. Os níveis das áreas-chave e a Matriz de Maturidade dão várias possibilidades para definir a melhoria gradual das etapas.
- Formular plano: o plano aborda as atividades necessárias para orientar o processo de mudança em uma determinada direção.
- Implementar ações de melhoria: execução do plano. São analisadas as ações executadas e bem sucedidas.
- Avaliação final: qual foi o rendimento das ações implementadas? Nesta fase, o objetivo é mensurar as ações que foram executadas com sucesso, bem como avaliar se as metas iniciais foram cumpridas. Com base nestas observações, a decisão sobre a continuação do processo de mudança será tomada.

10.4.2. TMM – Test Maturity Model

O *Testing Maturity Model* – TMM [Burnstein, et al 1998] foi desenvolvido pelo Illinois Institute of Technology como um guia para melhoria de processos de testes. A estrutura do TMM foi baseada no CMM, e está aderente ao CMMI, consistindo de cinco níveis que avaliam o grau de maturidade de um processo de testes. Para cada nível de maturidade áreas de processo são definidas. Uma área de processo é um conjunto de atividades que, quando executadas de forma adequada, contribuem para a melhoria do processo de testes. Na Figura 10.11 pode-se observar a estrutura do TMM.



Comment [j236]: itálico

Figura 10.11. Estrutura do TMM

10.4.2.1. Níveis de Maturidade do TMM

Os cinco de níveis de maturidade do TMM mostram uma evolução de um processo caótico e indefinido para um processo de testes controlado e otimizado. Nesta seção, será visto o detalhamento de todos os níveis de maturidade do TMM.

- Nível 1 – **Initial**

Os testes são caóticos, não existe processo definido. O objetivo dos testes é mostrar que o software funciona sem maiores falhas.

Áreas do processo: Nenhuma área do processo é identificada nesse nível;

- Nível 2 – **Phase Definition**

Ocorre a definição de um processo de teste. No contexto de estruturação do processo de testes, planos de teste são estabelecidos contendo estratégias de teste. Os testes ainda acontecem tardiamente dentro do ciclo de vida do desenvolvimento.

Áreas do processo: Políticas e objetivos, planejamentos dos testes, técnicas, métodos e ambiente de testes;

- Nível 3 – **Integration**

Os testes são completamente integrados ao ciclo de vida do software, sendo reconhecido em todos os níveis do processo de testes. O planejamento acontece no estágio inicial dos projetos, através de um Master Test Plan. A estratégia de testes é determinada através de técnicas de gerenciamento de riscos e baseada em requisitos. Programas de treinamento e revisões fazem parte do processo.

Áreas de processo: Organização dos testes, programa de treinamento, ciclo de vida, integração, controle e monitoramento;

- Nível 4 – **Management and Measurement**

Os testes são completamente definidos, bem fundamentados e medidos. Revisões e inspeções são incorporadas ao ciclo de vida do desenvolvimento e considerados parte dos testes. Os produtos de software são avaliados a partir de critérios de qualidade por características de qualidade, como reusabilidade, usabilidade e manutenibilidade. Casos de testes são armazenados e gerenciados em uma base de dados central para reuso e testes de regressão.

Comment [j237]: referencia

Comment [j238]: referencia

Comment [j239]: traduzir

Comment [j240]: traduzir

Comment [j241]: traduzir

Comment [j242]: traduzir

- **Nível 5 – Optimization**

Resultados são arquivados com o objetivo de melhoramentos em estágios anteriores, a área de testes é completamente definida através de seu processo e capaz de controlar seus custos, sendo efetivos. No nível 5 métodos e técnicas são otimizados e estão em melhoramento contínuo. A prevenção de defeitos e o controle de qualidade são introduzidos em outras áreas do processo. Há procedimentos para escolha e avaliação de ferramentas de testes. Testar é um processo com o objetivo de prevenir defeitos.

Áreas do processo: Prevenção de defeitos, controle de qualidade e otimização do processo de teste.

Os cinco de níveis de maturidade mostram uma evolução de um caótico e indefinido para um controlado e otimizado processo de testes.

Comment [j243]: traduzir

10.4.3. TIM – Test Improvement Model

Desenvolvido pela Ericson, Subotic e Ursing o TIM [Koomen & Pol 1999] foi concebido pelos desenvolvedores que sentiam a necessidade de melhorar o processo de testes. O TIM se propõe a identificar o estado atual das práticas das áreas chaves e serve como um guia na implementação dos pontos fortes e na remoção dos pontos fracos.

O TIM é composto de um Modelo de Maturidade e um Processo de Avaliação.

10.4.3.1. Modelo de Maturidade

Consiste em quatro níveis, o primeiro foi denominado de nível 0, considerado um nível de não conformidades, no entanto os outros níveis, de 1 a 4, possuem nomes os quais representam seus objetivos gerais e também sub-objetivos. Um objetivo só poderá ser atendido se seus sub-objetivos forem atendidos também. Outro componente do TIM são as áreas chaves. Existem cinco ao todo e cada uma delas cobre partes importantes da disciplina de teste. Outro fator importante é que as Áreas Chaves possuem os mesmos nomes dos Níveis

Os Níveis e seus sub-objetivos estão listados abaixo:

- **Nível 1 – Baselineing**

- Padronização dos documentos, métodos e políticas.
- Análise e classificação dos problemas.

- **Nível 2 – Cost-effectiveness**

- Detecção de bugs desde o início do projeto
- Automação de tarefas de teste
- Treinamento
- Reuso

Comment [j244]: colocar entre parentese e colocar tradução

Comment [j245]: referencia

Comment [j246]: traduzir

Comment [j247]: traduzir

Comment [j248]: traduzir

- Análise de custo x benefício para justificar os gastos
- Análise de problemas nos produtos e processos
- Métricas de produtos, processos e recursos
- Análise e gerenciamento de riscos
- Comunicação com todas as partes envolvidas nos projetos
- **Nível 4 - *Optimizing***
 - Conhecimento e entendimento através de experimentação e modelagem
 - Cooperação com todas as partes envolvidas nos projetos em todas as fases do desenvolvimento
 - Análise das causas raízes para os principais problemas
 - Melhoria contínua

Comment [j249]: traduzir

10.4.3.2. **Áreas Chave**

As Áreas Chaves do TIM [Koomen & Pol, 1999] estão listadas abaixo e para cada nível de maturidade são apresentados os principais aspectos da disciplina de teste, são eles:

Comment [j250]: traduzir subtopicos, que puderem ser traduzidos

- **Aspecto: Organização**
 - No nível *Baselining* deve-se organizar um conjunto mínimo de papéis para executar as atividades básicas de teste.
 - No nível *Cost-effectiveness* a principal mudança em relação ao modelo organizacional é a independência da equipe de teste.
 - No nível *Risk-lowering* aumenta a interação entre as equipes de desenvolvimento e a equipe de teste. A equipe de teste necessita conhecer mais sobre o desenvolvimento do produto, de forma a aumentar a qualidade do produto e o conhecimento das regras de negócio.
 - No nível *Optimizing* os testadores fazem parte do time de desenvolvimento e possuem conhecimento em várias disciplinas. São estabelecidos grupos com o objetivo de avaliar continuamente o processo.
- **Aspecto: Planejamento e Rastreabilidade**
 - No nível *Baselining* o projeto de teste possui um planejamento básico, nele são estabelecidos critérios de entrada e saída, os resultados dos testes são documentados, processados e distribuídos.
 - No nível *Cost-effectiveness*, o planejamento e a rastreabilidade são auxiliados por ferramentas, alguns planos genéricos são utilizados. A escolha dos estágios e métodos de testes é alinhada de acordo com os objetos e os objetivos dos testes.

- No nível *Risk lowering*, a análise dos riscos é realizada e sua influência é bastante elevada no planejamento, além de afetar partes do plano, mais precisamente os objetivos dos testes.
 - No nível *Optimizing*, atividades de planejamento e a rastreabilidade é continuamente melhorada baseada na análise de métricas. Reuniões de *post-mortem*¹⁴ são realizadas e os resultados armazenados e distribuídos.
 - **Aspecto: Casos de Teste**
 - No nível *Baselining* os casos de testes são elaborados baseados nos requisitos de sistemas e segundo as instruções das políticas.
 - No nível *Cost-effectiveness*, os casos de testes são projetados de acordo com técnicas documentadas. Com armazenamento dos casos de testes a reusabilidade se torna possível.
 - No nível *Risk-lowering*, com o armazenamento dos casos de testes no nível anterior é possível selecioná-los de acordo com a criticidade.
 - No nível *Optimizing*, medições, revisões e melhorias são realizadas sobre os casos de testes.
 - **Aspecto: Testware (artefatos de teste)**
 - No nível *Baselining* os problemas são reportados e computados.
 - O nível *Cost-effectiveness* se caracteriza pelo uso de ferramentas de cobertura, banco de dados para gerenciar o *testware*.
 - No nível *Risk-lowering*, são realizados testes de regressão quando o código sofre alteração. A análise de risco é realizada com uso de ferramentas.
 - No nível *Optimizing* Ambiente de Teste é Integrado.
 - **Aspecto: Revisões**
 - No nível *Baselining*, Padrões de revisões de documentos são utilizados.
 - No nível *Cost-effectiveness*, os projetos e códigos são documentados e revisados através de técnicas de revisão escolhidas pela organização.
 - Técnicas de revisão e inspeção são constantemente evoluídas. Todo o *testware*, o processo e produto são revisados e medidos no nível de *Risk lowering*.
-

- No nível *Optimizing*, técnicas e time são selecionados baseados em fatos.

Através dos níveis de maturidade e áreas chaves o TIM é possível avaliar um processo de teste.

Considerações Finais

O desenvolvimento de software engloba um mercado de extrema competitividade. Tendo em vista que os sistemas que apresentam melhor qualidade garantem seu espaço no mercado, as empresas que os desenvolvem têm investido bastante para assegurar a qualidade de seus produtos e garantir a satisfação dos clientes. A qualidade de um produto pode ser definida como sua capacidade de cumprir os requisitos inicialmente estipulados pelos clientes, e sendo assim, está diretamente relacionada à qualidade do processo de desenvolvimento. Por este motivo, tem surgido uma grande demanda ao incentivo de pesquisas que levem em consideração à procura por formas de melhoria da qualidade dos produtos.

Este capítulo procurou introduzir ao leitor boas práticas no que diz respeito à qualidade dos produtos, apresentando um conjunto de normas que representam a padronização mundial para avaliação da qualidade de produtos de software. As atividades de teste e inspeção também foram destacadas como forma de encontrar defeitos no software e corrigi-los, antes de entregar o produto a seus clientes, e analisar se o sistema faz o que é suposto fazer. Finalmente, modelos de maturidade de testes foram apresentados como mais uma tentativa de alcançar melhorias na qualidade do processo de teste de software, que afeta diretamente a qualidade do produto.

Comment [j251]: aumentam as possibilidades, mas nao garantem..

Exercícios

1. Quais são as diretrizes para uso da norma NBR ISO/IEC 9126-1?
2. A que se propõe a norma ISO 12119?
3. Que subdivisões da norma ISO 14598 estabelecem itens necessários para o suporte à avaliação?
4. Quais são os componentes do projeto SQuaRE? Defina-os.
5. Qual a diferença entre testes e inspeções de software?
6. Cite 5 tipos de testes e explique cada um deles.
7. Quais os estágios de testes possíveis e quais as características de cada um deles?
8. O que são testes beta?
9. O que são testes de regressão?
10. Qual a diferença entre a abordagem de caixa preta e a abordagem de caixa branca?
11. Quais são os papéis existentes na equipe de inspeção de software e quais suas responsabilidades?
12. Quais são as etapas do processo de inspeção de software? Explique cada uma delas.
13. Explique como é feita a implantação da melhoria no TPI.
14. Defina os níveis de maturidade do TMM.
15. No aspecto Organização, como são caracterizados os níveis de maturidade do TIM?

Sugestões de Leitura

Para conhecer mais sobre normas de qualidade de produto de software, é recomendada a leitura do livro *Tecnologia da Informação: Qualidade de Produto de Software*, Guerra & Colombo 2009.

Para ampliar o entendimento sobre o assunto de teste de software é recomendada a leitura do livro *Foundations of software testing*, Graham, D., Veenendaal, E. v., Evans, I. and Black, R., 2007. Este livro é utilizado por pessoas que desejam tirar o certificado ISTQB (*International Software Testing Qualifications Board*), portanto, é muito interessante para adquirir melhores conhecimentos sobre este conteúdo.

Para um melhor conhecimento sobre os conceitos e o processo de inspeção de software é sugerida a leitura de *Design and Code Inspection to Reduce Errors in Program Development*, Fagan, M.E.,1976.

Para se aprofundar mais sobre as ferramentas de inspeção de software é recomendada a leitura de *Modern Software Review Techniques and Technologies*, Wong, Y. K., 2006.

Para melhor conhecimento sobre o TPI (*Test Process Improvement*) é recomendada a leitura do livro *Test Process Improvement A practical step-by-step guide to structured testing*, Koomen & Pol, 1999.

Para aprofundar a leitura sobre TMM (*Test Maturity Model*), é sugerida a leitura do livro *A Model to Assess Testing Process Maturity*, Burnstein & Grom, 1998.

Tópicos de Pesquisa

Existem vários estudos atualmente na academia no que diz respeito à seleção de testes de regressão, uma vez que executar todos os casos de teste novamente sempre que uma nova versão do sistema for liberada é uma prática inviável. Dessa forma, várias pesquisas e propostas de soluções e técnicas para realizar uma quantidade suficiente de testes que atinja a cobertura necessária para garantir a corretude do software podem ser encontradas na literatura.

Outra área de pesquisa bastante desafiadora na área de teste de software é a geração automática de casos de teste, considerando que a elaboração de casos testes manualmente é um processo que consome muito tempo e esforço. Sendo assim, diversas propostas são elaboradas dia após dia com o objetivo de tornar o processo de teste mais ágil, menos suscetível a erros e dependente da interação humana.

Na área de inspeção de software, grandes desafios podem ser observados com o objetivo de encontrar estratégias para diminuir a quantidade de defeitos de um software. Na literatura, podem ser encontradas pesquisas e artigos com estudos focados neste objetivo.

Na área de modelo de maturidade de testes, há uma organização, chamada *TMMi Foundation*, sem fins lucrativos, em Dublin – Irlanda, que foi fundada para tentar transformar o modelo TMM em uma norma e, conseqüentemente, promover a sua aceitação como um padrão da indústria internacional de avaliação e de organizações de teste de software. A Fundação TMMi tem como objetivo: a criação e gestão de uma organização independente, imparcial com repositório central de dados e prestação de serviços, métodos de avaliação com base no modelo padrão, definição e manutenção de avaliadores independentes e prestação de um fórum público das partes interessadas para facilitar a livre troca de informação, educação, idéias e uso da norma pública.

Em relação ao TPI, há pesquisas na academia que objetivam melhorar a produtividade do time de testes, utilizando as práticas definidas pela melhoria gradual do processo de testes.

Comment [j252]: acredito que esse não seja o formato de tópicos de pesquisa, dúvida!

Referências

Comment [j253]: padronizar

Cortes, M. L. (2009). *Qualidade*. Acessado em 03 de Setembro de 2009, disponível em <http://www.ic.unicamp.br/~cortes/mc726/cap3.pdf>

Vera, P. C. (2009). *Qualidade*. Acessado em 05 de Setembro de 2009, disponível em <http://inf.unisul.br/~vera/egs/egs02.htm>

Burnstein, I., Homyen, A., Grom, R. and Carlson C.R. (1998), "A Model to Assess Testing Process Maturity", Crosstalk

Koomen, T., and Pol, M. (1999). *Test Process Improvement A practical step-by-step guide to structured testing*. ACM Press.

Boehm, B. W. and BASILI, V.R. (2001) "Software Defect Reduction Top 10 List.", *IEEE Computer* 34 (1), p. 135-137.

Pressman, R. S., *Engenharia de Software*, McGraw Hill, 2002.

Sommerville, I., *Software Engineering*, 7th Edition, Addison Wesley, 2004.

Guerra, A., C and Colombo, R., M., T. (2009) "Tecnologia da Informação: Qualidade de Produto de Software", PBQP Software.

Fagan, M.E. (1976) "Design and Code Inspection to Reduce Errors in Program Development", *IBM Systems Journal*, vol. 15, no. 3, p. 182-211.

Hedberg, H. (2004) "Introducing the Next Generation of Software Inspection Tools", In: *International Conference of product focused software process improvement*, 5, Kansai. *Lecture notes in computer science*, Berlin: Springer, p. 234-247.

Graham, D., Veenendaal, E. v., Evans, I. and Black, R. "Foundations of software testing", *ISTQB Certification*, Thomson Learning, 2007.

Wong, Y. K. "Modern Software Review Techniques and Technologies", IRM Press, 2006.

Selby, R. W. and Basili, V. R., et al. (1987). "Cleanroom software development: an empirical evaluation", *IEEE Trans on Software Engineering* , SE-13(9), 102-37. (Chs. 4,22).

Mills, H. D. and Dyer, M., et al. (1987). "Cleanroom software engineering", *IEEE Software*, 4(5), 19-25. (Chs. 3,4,22).

Zeilinger, C., (2003). "Robustness of Software", *Seminar Software-Development (programming styles)*.

Myers, G. J., *The Art of Software Testing*, 2nd Edition, John Wiley & sons, Inc, 2004.

Capítulo

11

Uma introdução ao SWEBOK 2004

André Luís de L. Torres

Nas últimas décadas, a computação tem se desdobrado em uma extensa lista de subáreas de estudo. A quantidade de informação aumentou de tal modo que a especialização profissional tornou-se comum, de modo a alcançar o nível de excelência desejado. Este capítulo tem o objetivo de apresentar o SWEBOK (Software Engineering Body of Knowledge), um guia para o corpo de conhecimento em Engenharia de Software, patrocinado pelo IEEE (Institute of Electrical and Electronics Engineering).

11.1. Introdução

O termo Engenharia de Software foi usado pela primeira vez? como tema na conferência da OTAN no ano de 1968, evento esse que foi motivado pela crise no desenvolvimento de software. Nesta época, a demanda por métodos padronizados para a qualidade de software era muito grande, visto que a maioria das aplicações foi produzida sem nenhuma padronização. John Ronald Graham comentou "construímos sistemas como os irmãos Wright construíam aviões - constrói-se de uma só vez, empurra-se para o despenhadeiro, deixa bater e começa tudo outra vez" [NAUR & RANDELL, 1968].

Comment [F254]: Não deve haver espaço para o primeiro parágrafo.

O aumento gradual e crescente da capacidade de processamento dos computadores revelou a necessidade de se criar processos que orientassem e organizassem a atividade de desenvolvimento de software, deixando de ser uma atividade que até então supria apenas as necessidades do hardware.

Desde os primeiros computadores comerciais, os softwares implantados ou lançados no mercado se caracterizam, na sua maioria, pela presença de erros encontrados nas fases de verificação e validação, como por exemplo: erros em estimativas, dificuldade no domínio da área de conhecimento específica do software proposto, especificações obscuras, requisitos mal feitos e mal interpretados, conflitos nos objetivos e mudanças intermináveis e mal controladas.

Tais erros aumentaram a importância e responsabilidades dos especialistas ligados a uma das áreas da computação, conhecida como Engenharia de Software. Com isso, a *Institute of Electrical and Electronics Engineers* (IEEE) e a *Association for Computing Machinery* (ACM) conduziram estudos de modo a promover ativamente a Engenharia de Software como uma profissão desde 1993, definindo as fronteiras que delimitam a Engenharia de Software, através do Corpo de Conhecimento em Engenharia de Software - *Software Engineering Body of Knowledge* (SWEBOK). (SWEBOK, 2004).

Neste capítulo iremos apresentar uma descrição sobre as áreas do SWEBOK e suas idéias gerais, que embasam, através do seu guia, profissionais, sociedade científica e órgãos públicos para o conhecimento da Engenharia de Software.

11.2. O Projeto do SWEBOK

O SWEBOK surgiu através de uma parceria entre a IEEE, a Computer Society e ACM a fim de promover a profissionalização da Engenharia de Software e criar um consenso sobre as áreas de conhecimento da Engenharia de Software e seu escopo. Sendo iniciado em 1998 pelo Software Engineering Coordinating Committee (SWECC) e financiado por organizações como a ACM, a Boeing, o Conselho Canadense de Engenheiros Profissionais, Construx Software, MITRE Corporation, entre outras.

Comment [F255]: Espaço do parágrafo

O SWEBOK é recomendado para diversos tipos de público, em todo o mundo, com o objetivo de ajudar organizações a terem uma visão consistente da Engenharia de Software. É endereçado a gerentes, engenheiros de software, às sociedades profissionais, estudantes, professores e instrutores desta área de conhecimento.

Os objetivos do SWEBOK são:

- Oferecer uma visão consistente da Engenharia de Software no âmbito mundial;

- Deixar claros os limites da Engenharia de Software com respeito a outras disciplinas como ciência da computação, gerência de projetos, engenharia da computação, matemática, entre outros;
- Caracterizar o conteúdo da disciplina de Engenharia de Software;
- Prover acesso aos tópicos do corpo de conhecimento da Engenharia de Software;
- Prover uma base para desenvolvimento curricular e para certificação individual
- Como material de apoio.

11.2.1. Categorias do Conhecimento da Engenharia de Software

São consideradas três bases de conhecimento para categorizar o guia SWEBOK, conforme a tabela 11.1.

Comment [F256]: Faltou um verbo. Ex: Servir como material de apoio.

Especializado Práticas usadas apenas por alguns tipos de software	Geralmente Aceitas Práticas tradicionais estabelecidas recomendadas pela maior parte das organizações
	Pesquisa Avançada Práticas inovadoras usadas apenas por algumas organizações com conceitos a serem desenvolvidos e testados em organizações de pesquisa.

Comment [F257]: Essas fontes estão muito grandes. Ver a formatação padrão de tabelas.

Tabela 11.1. Categorias do Conhecimento conforme o SWEBOK

O guia, como é convencionalmente chamado, divide a Engenharia de Software em onze áreas de Conhecimento - *Knowledge Areas* (KAs): requisitos, gerência de engenharia, projeto, métodos e ferramentas de engenharia, construção, processo de engenharia, testes, qualidade, manutenção, disciplinas relacionadas à gerência de configuração, as quais serão explicadas nas subseções a seguir. É importante comentar que o IEEE oferece duas modalidades de certificação sobre o SWEBOK, disponibilizadas para engenheiros e desenvolvedores, são elas:

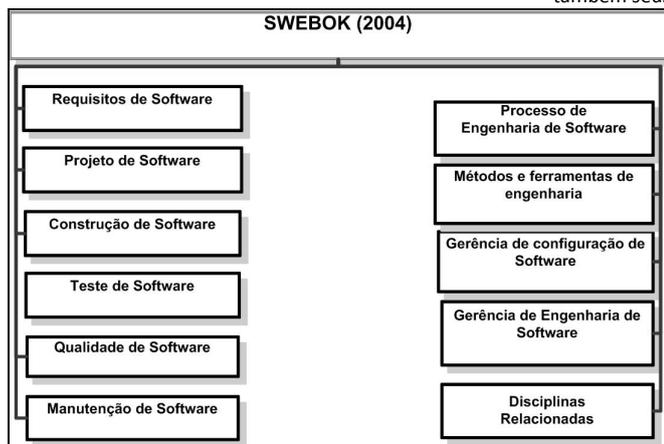
- **Certificação de associados para o desenvolvimento de Software - *Certified Software Development Associate* (CSDA);**
- **Certificação Profissional para desenvolvimento de Software - *Certified Software Development Professional* (CSDP).**

Depois de pesquisa extensa, os líderes da IEEE reconheceram a necessidade por de um endereçamento do programa de certificação aos iniciantes e profissionais no desenvolvimento de software. O propósito do programa de CSDA é estabelecer um nível mínimo de habilidades e conhecimento, sendo o primeiro passo para se tornar um Profissional de Desenvolvimento de Software Certificado (CSDP).

A certificação CSDP é recomendada para profissionais mais experientes e tem como objetivo ampliar as habilidades e conhecimentos técnicos especializados relevantes sobre o SWEBOK.

11.2.2. Áreas de Conhecimento

O SWEBOK está organizado em dez áreas de conhecimento e um tópico referente às disciplinas relacionadas com a Engenharia de Software. A figura 11.1 apresenta o corpo de conhecimento do guia, como também seus níveis hierárquicos.



Fonte: Adaptado do SWEBOK,2004.

Figura 11.1. Organização do SWEBOK.

A seguir serão descritas cada uma das áreas de conhecimento do SWEBOK.

Requisitos de Software

Os requisitos expressam a necessidade e restrições colocadas sobre o produto de software que contribuem para a solução de algum problema do mundo real. Esta área envolve elicitação, análise, especificação e validação dos requisitos de software [SWEBOK, 2004].

As principais falhas em projetos são relativas aos requisitos, devido as às dificuldades no entendimento das necessidades do usuário. Portanto, realizar corretamente o levantamento e administração de requisitos é vital para a qualidade de software [KOSCIANSKI E SOARES, 2007].

A área de requisitos de software está dividida em sete sub-áreas:

- a. **Fundamentos dos Requisitos**

Comment [F258]: O nome da tabela deve ficar acima.

Comment [F259]: Não são 10? Requisitos, Projeto, Construção, Testes, Manutenção, Configuração, Gerenciamento, Processo, Ferramentas e Métodos, Qualidade.

Comment [F260]: Seria bom numerar cada uma pra ficar mais explícito.

Comment [F261]: Verificar formatação.

Formatted: Font: Not Bold

O SWEBOOK define requisitos como uma propriedade que deve ser observada a fim de resolver algum problema do mundo real. Logo, o problema pode ser para automatizar parte de uma tarefa a ser realizada por alguém na utilização de um software.

Esta sub-área inclui as definições dos próprios requisitos de software e também dos tipos principais de requisitos: de produto, de processo, funcionais, não-funcionais e emergentes (dependem de vários componentes simultaneamente). Esta sub-área também descreve a importância de requisitos quantitativos e distingue-os entre requisitos de sistema e requisitos de software.

b. Processo de Requisitos

O processo de requisitos de software orienta o planejamento de requisitos, de forma de efetuar o entendimento das outras cinco sub-áreas com o processo completo de planejamento de software. Esta sub-área mostra como os processos de requisitos se integram com os processos de engenharia de software sendo dividida em quatro sub-áreas: descreve 1) Modelos de processos, 2) Atores dos processos, processos de suporte e gerência do processo 3) Gerenciamento e suporte aos processos e 4) Processos de qualidade., além de tratar da qualidade e da melhoria do processo.

c. Declaração Elicitação de Requisitos

A declaração elicitação de requisitos preocupa-se com a coleta dos requisitos de software pelo engenheiro de software. Identifica as fontes dos requisitos e define as técnicas para extraí-los os requisitos. É o primeiro estágio para o entendimento do problema disposto. Está dividida em duas sub-áreas: 1) Fontes de requisitos e 2) Técnicas de elicitação.

d. Análise de Requisitos

Alguns cuidados devem ser tomados para descrever com precisão os requisitos. Esta sub-área preocupa-se em detectar e resolver conflitos entre requisitos, descobrir as fronteiras do software e como ele interagirá com o ambiente, além de aprimorar os requisitos do sistema para derivar requisitos de software. Está dividida em quatro sub-áreas: 1) Classificação dos requisitos, 2) Modelagem conceitual, 3) Projeto arquitetural e Distribuição de requisitos e 4) Negociação de requisitos.

A análise de requisitos é uma tarefa da engenharia de software que efetua a ligação entre a alocação de software em nível de sistema e o projeto de software, possibilitando que o engenheiro aprimore e construa modelos do processo, dos dados e dos domínios comportamentais que serão tratados pelo software. [PRESSMAN, 1995].

(Pressman,2002) cita que “se você não analisa, é altamente provável que construa uma solução de software muito elegante que resolve o problema errado”. Esta atitude pode resultar em perda de tempo e dinheiro, pessoas frustradas e clientes insatisfeitos, conforme ilustrado na figura 11.2.

Comment [F262]: Qual o modelo padrão de referência?

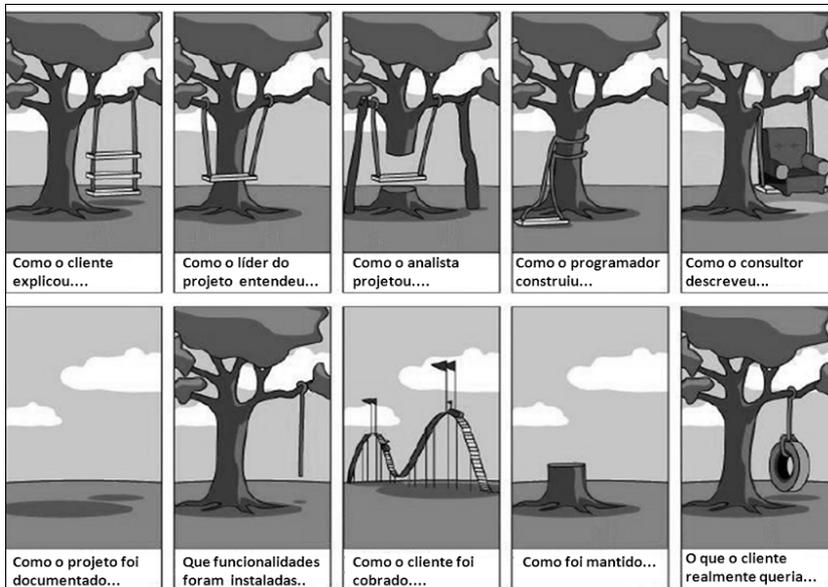


Figura 11.2. Análise de requisitos

Formatted: Font: Not Bold

e. Especificação de Requisitos

Para muitos profissionais da engenharia, conforme o guia, especificação refere-se à atribuição de valores numéricos ou limites para os objetivos do projeto. No entanto, a principal atividade desta fase é a confecção da documentação do sistema, que pode ser sistematicamente revisado, validado e aprovado, especificando os componentes de software. Está dividida em quatro sub-áreas: 1) Documento de definição do sistema, 2) Especificação dos requisitos de sistemas e 3) Especificação dos requisitos de software.

f. Validação de Requisitos

A documentação de requisitos pode ser objeto de validação. Tal atividade busca a conformidade do documento com os padrões da organização. Possui as etapas de 1) rRevisão dos requisitos, 2)pPrototipação, 3) Validação de modelos e 4)t Testes de aceitação.

g. Considerações Práticas

Esta sub-área descreve os tópicos que necessitam ser compreendidos na prática, validando os atributos dos requisitos como também tem o papel de avaliar o tamanho das mudanças nos requisitos e estimar os custos do desenvolvimento e manutenção.

O primeiro tópico é a natureza iterativa do processo dos requisitos. Os três tópicos seguintes são fundamentalmente sobre gerência da mudança e a manutenção dos requisitos em um estado que espelhe exatamente o software a ser construído. Inclui a gerência da mudança, os atributos dos requisitos e a seqüência rastreabilidade dos requisitos. O tópico final é sobre processos de medição dos requisitos.

As principais responsabilidades das sub-áreas de requisitos de software são:

- Preocupação com a origem dos requisitos e como os engenheiros de software podem coletá-los. Inclui fontes e técnicas de levantamento de requisito.
- Descrever a importância dos requisitos quantificáveis, e distinguir entre sistemas e requisitos de software.
- Demonstração de como o planejamento de requisitos se encaixa com o processo completo de planejamento de software.

- Preocupação com os modelos de processo, atores, suporte, gerenciamento de requisitos, melhoria e qualidade do processo.

Projeto de software

Projeto de software ou Design de Software é a atividade do ciclo de vida da Engenharia de Software em que os requisitos do software são analisados a fim de produzir uma descrição da estrutura interna do software que servirá como base para sua construção [SWEBOK, 2004].

Esta é uma área de grande importância, pois compreende tanto a definição da arquitetura, componentes, relações, e outras características de um sistema ou de um componente quanto o resultado do próprio processo [IEEE, 1990].

A área de projetos de software está dividida em sete seis sub-áreas:

a. Fundamentos do Projeto de Software

São os conceitos, noções e terminologias introduzidas como base fundamental para a compreensão do papel do projeto de software.

b. Questões Chave no Design de Software

Trata dos assuntos que devem ser abordados no projeto de software, tais como: controle e tratamento de eventos, concorrência, tratamentos de erros e de exceções, entre outros.

c. Estrutura e Arquitetura de Software

Descreve a estrutura, estilo, padrões e frameworks utilizados para a arquitetura funcional do software.

d. Evolução e Análise de Qualidade do Projeto de Software

Descreve tópicos relacionados com a qualidade de software, como: métricas, avaliação de ferramentas e características de qualidade.

e. Notações do Projeto de Software

Apresenta notações estruturais (estáticas) e comportamentais (dinâmicas).

f. Estratégias e Métodos para o Projeto de Software

Descrição de método para o projeto de software, bem como de um conjunto de orientações na utilização de tais métodos. São eles: métodos orientados a funções, objetos, formais e transformacionais.

Construção de software

Refere-se à criação do conjunto de programas (componentes) que compõem o software através de uma combinação de codificação, verificação, testes unitários, testes de integração e depuração.

A construção de software está ligada a todas as outras KA's, mais fortemente ao Projeto de Software e Teste de Software. Isso ocorre porque o processo de construção do próprio design de software envolve vários testes das suas atividades. (SWEBOK, 2004).

As áreas correlatas à construção de software, segundo o SWEBOK (2004) são:

a. Fundamentos da Construção de Software

Tem como objetivos principais minimizar a complexidade, antecipar as mudanças, efetuar a verificação e definir os padrões para a construção de software.

b. Gerenciamento da Construção de Software

Efetua o planejamento e avaliação da construção do software, como também informa os modelos para tal atividade. Exemplos dos modelos: Prototipação, Cascata, eEspiral, etc.

c. Considerações Práticas da Construção de Software

Nessa sub-área são descritas atividades práticas como: Projeto de Construção, Linguagem Própria, Codificação, Teste, Construção, Reuso, Qualidade, Integração.

O guia SWEBOK recomenda que nesta etapa as funcionalidades do software sejam testadas durante todo o processo de desenvolvimento, não deixando apenas para a etapa de testes.

Teste de software

Nas décadas de 60 e 70, os desenvolvedores dedicavam a maior parcela dos seus esforços nas atividades de codificação e nos testes unitários. Sendo apenas uma parcela menor dedicada à integração dos programas e nos testes dos sistemas, pois as atividades de testes eram consideradas um mal necessário e não eram tratadas como um processo formal alinhado ao processo de desenvolvimento dos sistemas. (RIOS E MOREIRA, 2006).

Nessa seção é feito apenas um breve comentário sobre a área de conhecimento de teste de software conforme o SWEBOK, devido ao seu embasamento teórico [já esata](#) está detalhado no capítulo 10 desse livro.

O teste é uma atividade realizada para avaliação da qualidade do produto, efetuando sua melhoria através da identificação de defeitos e problemas (SWEBOK, 2004).

O destaque crescente do software como elemento de sistema e os “custos” envolvidos associados às falhas de software são forças propulsoras para uma atividade de teste cuidadosa e bem planejada (PRESSMAN, 1995).

Essa área descrita no SWEBOK consiste na verificação dinâmica do comportamento de um programa com um conjunto finito de casos de testes, selecionados de um domínio geralmente infinito de execuções, para confirmar o comportamento especificado esperado.

São sub-áreas dessa área de conhecimento: [f](#)undamentos [do teste de software](#), [n](#)íveis de Teste de Software, [T](#)técnicas [de teste de software](#), [m](#)edidas [de teste de software](#), [p](#)rocesso [de teste](#) e [c](#)onsiderações práticas.

É atualmente considerado [Considera-se atualmente](#) que a atitude certa para qualidade é a de prevenção, sendo muito melhor evitar problemas do que corrigi-los. Mas ao se debater com falhas em software já concluído, um bom plano de manutenção corretiva é um remédio a ser planejado pela equipe. Na próxima seção é abordada a manutenção de software como uma estratégia descrita no SWEBOK para busca [contínua](#) da qualidade de software.

Manutenção de software

O desenvolvimento de software requer esforços que visam a entrega de um produto que satisfaça [aos](#) requisitos do cliente. Nesta área de conhecimento sua principal responsabilidade é totalizar as atividades requeridas para fornecer suporte custo-efetivo a um sistema de software, que pode ocorrer antes ou depois da entrega. Antes da entrega do software são realizadas atividades de planejamento e depois, modificações são feitas com o objetivo de corrigir falhas, melhorar o desempenho ou adaptá-las a um ambiente externo (SWEBOK, 2004).

No guia, a manutenção de software está relacionada [ao](#) com todos os aspectos da Engenharia de Software, sendo assim, ligada [a](#) [todas](#) as áreas do SWEBOK. Sendo seu objetivo principal sustentar o produto ao longo do seu ciclo de vida operacional.

O fluxo de eventos que podem ocorrer como resultado de um pedido de manutenção é ilustrado na figura 11.3. Onde, (Pressman 2002) [se](#) o único elemento disponível de uma configuração de software for o código-fonte, a atividade de manutenção inicia-se com uma penosa avaliação do código, etapa essa dificultada pela [freqüente](#) documentação interna [estará](#) num estado ruim.

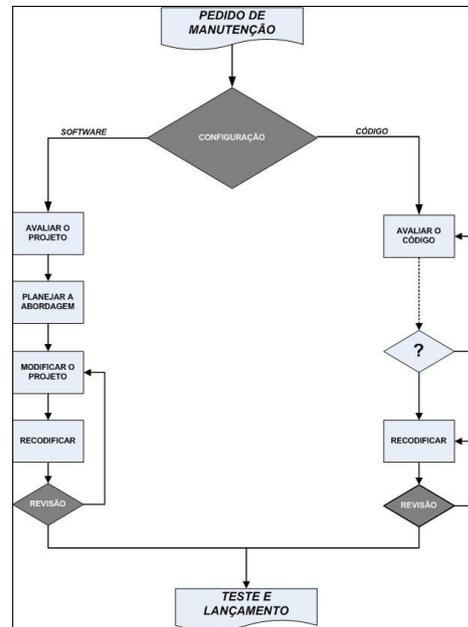


Figura 11.3. Manutenção estruturada *versus* não estruturada

As áreas correlatas à manutenção de software, segundo o SWEBOK (2004) são:

a. Fundamentos da Manutenção de Software

Apresenta os conceitos e terminologias que formam a base de conhecimento para compreensão do papel da manutenção de software, como por exemplo, a natureza, categorias, custos, evolução de software, entre outras.

b. Questões Chave sobre Manutenção de Software

Apresenta as questões chave relacionadas com a manutenção de software, agrupadas como: problemas técnicos (compreensão limitada, teste, análise de impactos e manutenção), questões de gestão (alinhamento com os objetivos organizacionais, processo, aspectos organizações de manutenção e outsourcing), estimativas de custos e medidas (parâmetros e experiência).

c. Processo de Manutenção de Software

Fornece referências e padrões utilizados para implementar a manutenção de software, relacionando com outras atividades da Engenharia de Software. A figura 11.4 ilustra o processo de manutenção de software conforme a ISO/IEC 14764, que é similar a IEEE 1219 (atividades de manutenção de software).

Formatted: Font: Not Bold

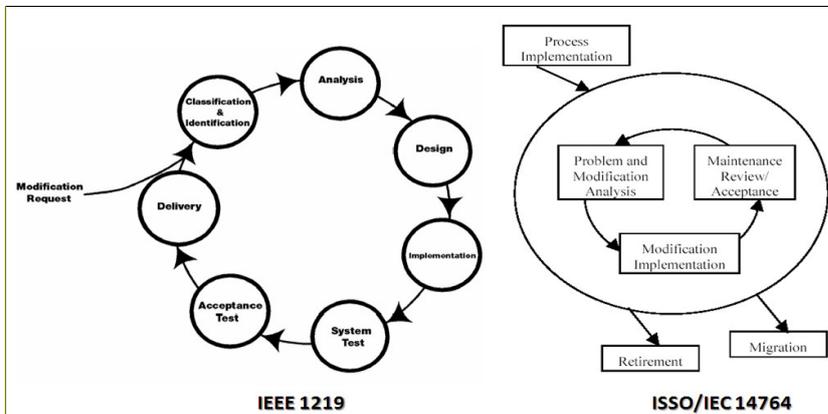


Figura 11.4. Atividades do processo de manutenção versus Processo de Manutenção de software

Comment [F263]: Essa figura não está muito legível, teria como refazê-la para ficar mais claro?

d. Técnicas de Manutenção de Software

Descreve técnicas de manutenção, por exemplo, compreensão do programa, reengenharia, engenharia reversa.

Em última análise, algumas organizações de software podem permanecer voltadas à manutenção, incapazes de embarcar em novos projetos, porque todos os seus recursos são dedicados à manutenção de velhos programas.

Gerência de configuração de Software

A Gerência de Configuração de Software – *Software Configuration Management (SCM)* é a disciplina que identifica a configuração de um sistema em pontos distintos no tempo com a finalidade de controlar sistematicamente as mudanças para configurar e manter a integridade e rastreabilidade de todos os artefatos ao longo do ciclo de vida do sistema (SWEBOK, 2004).

São atividades da SCM:

a. Processo de Gerenciamento de Configurações

Controla a evolução e integridade de um produto. Do ponto de vista do engenheiro de software, uma implementação bem sucedida exige um SCM cuidadoso, sendo preciso o seu processo.

b. Identificação da Configuração

Esta atividade tem papel de controlar e administrar itens de configuração de software (Informação que é criada como parte do processo de Engenharia de Software), cada um deve ser nomeado separadamente. Essa atividade constitui a base para as outras atividades do SCM. Suas principais atividades são: identificação dos itens a serem controlados (configuração de software, itens de configuração do software, relacionamento entre os itens de configuração, versões de software, **B**aseline e aquisição dos itens de configuração de software).

c. Controle da Configuração

Abrange a gestão durante o ciclo de vida do software, como por exemplo: processo para determinar quais as mudanças a serem feitas, a autoridade para aprovar algumas alterações, apoio a implementação dessas mudanças, entre outras atividades. Uma Ordem de **M**mudança de **E**ngenharia – Engineering Change Order (ECO) é gerada para cada mudança aprovada. A ECO descreve as mudanças a serem feitas, as restrições que devem ser respeitadas e os critérios de revisão e auditoria. Informações obtidas a partir dessas atividades são úteis para a medição do tráfego da mudança e dos aspectos do retrabalho.

d. Registro do Estado da Configuração

É o registro e comunicação das informações necessárias para uma gestão eficaz da configuração de software. Desempenha um papel vital no sucesso de um grande projeto de desenvolvimento de software. São perguntas importantes dessa fase: o que aconteceu? Quem o fez? Quando aconteceu? O que mais será afetado?

e. Auditoria da Configuração

Formatted: Font: Not Bold

Formatted: Font: Not Bold

O SWEBOK detalha essa atividade baseado na norma IEEE 1028 e tem o objetivo de avaliar a conformidade dos produtos de software e processos de regulamento. As auditorias são conduzidas pelo processo que consiste na descrição dos papéis, contas e responsabilidades. Conforme (Pressman 1995), uma auditoria de configuração de software pergunta e responde às seguintes questões:

- A mudança especificada na ECO foi feita? Outras modificações adicionais foram incorporadas?
- Uma revisão técnica formal foi realizada para avaliar a exatidão técnica?
- Os padrões de Engenharia de Software foram adequadamente seguidos?
- A mudança foi “realçada” no SCI? A data da mudança e o autor da mudança foram especificados? Os atributos do objeto de configuração refletem a mudança?
- Os procedimentos de SCM para anotar a mudança, registrá-la e relatá-la foi seguido?
- Todos os SCIs relacionados foram adequadamente atualizados?

f. Entrega e Gerenciamento de Versões

A liberação é utilizada neste contexto para se referir a entrega de um item de configuração, incluindo a liberação interna. Quando as versões de diferentes itens de software estão disponíveis para entrega é frequentemente necessário para recriar versões específicas de pacotes e os materiais corretos para entrega da versão.

A figura 11. 5 ilustra as atividades citadas como também o seu relacionamento.

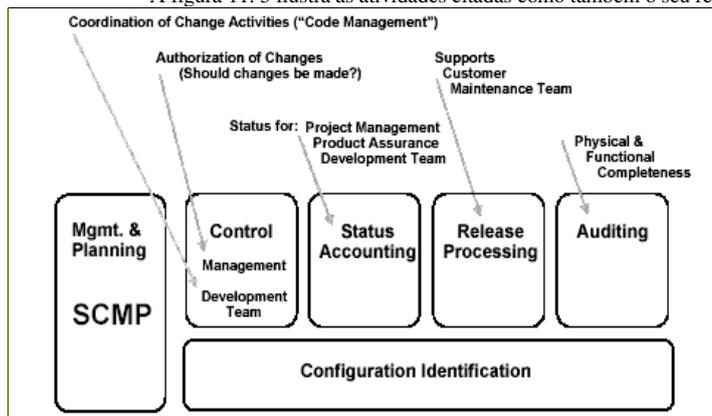


Figura 11.5. Atividades da SCM

É importante fazer uma distinção entre manutenção de software e gerenciamento de configuração de software. A manutenção é um conjunto de atividades de Engenharia de Software que acontece depois da entrega do software e colocado em operação, diferentemente do gerenciamento de configuração de software, que já efetua o controle e rastreamento do projeto de desenvolvimento de software desde o início, terminando quando o software é tirado de operação.

Gerência de Engenharia de Software

A área de “Gerenciamento de software” é uma área que influencia e recebe influência das outras áreas da Engenharia de Software, sendo desta forma fundamental para atingir um bom resultado ao final do projeto. É definida como a aplicação da gestão de atividades de planejamento, coordenação, medição, monitoramento, controle e comunicação de forma a garantir a manutenção sistemática e desenvolvimento dos projetos de software (SWEBOK, 2004).

No que se diz respeito à Engenharia de Software, o gerenciamento de software ocorre em três níveis: gerenciamento organizacional, gerenciamento de projeto e controle e planejamento de programas de medição. Sendo apenas, os dois últimos abordados em detalhes nesta seção.

Formatted: Justified, Indent: Left: 0,34", Space Before: 6 pt, Bulleted + Level: 1 + Aligned at: 0,74" + Indent at: 0,99", Tab stops: 0,5", Left

Comment [F264]: Essa figura deveria ser refeita em português.

Formatted: Font: Not Bold

Outro aspecto importante do gerenciamento são as atividades relacionadas com o gerenciamento de pessoas, levando em consideração equipe, clientes e responsáveis da própria organização. Tendo como responsabilidades o treinamento, motivação pessoal. Além disso, são necessários, os planejamentos e gerenciamentos da comunicação entre essas pessoas para melhorar o entendimento de todos, fator primordial para um bom resultado final.

Comment [F265]: Frase muito curta

A área de “Gerenciamento da Engenharia de Software” se divide em 7 sete sub-áreas:

a. Iniciação e Definição de Escopo

O foco é determinar os requisitos através de vários métodos de levantamento, avaliação do projeto e especificação dos requisitos e procedimentos para validação de mudanças. O guia destaca os seguintes tópicos relacionados a esta subárea: determinação e negociação dos requisitos, análise de viabilidade, revisão dos requisitos e do processo de software.

b. Planejamento do Projeto

Neste ponto, o ciclo de vida dos processos de software é avaliado para o melhor planejamento, levando em conta fatores como: natureza do projeto, complexidade funcional e técnica, requisitos de qualidade, entre outros. Tal sub-área é um fator importante para o sucesso da qualidade de software, detalhada na seção 11.2.2.11. Os tópicos relacionados são: planejamento do processo, esforço, previsão de custo, alocação de recursos, gerenciamento de risco e da qualidade.

c. Declaração do Plano de Projeto

A declaração do plano de projeto tem o papel de descrever o plano a ser implementado e os processos a ele incorporados, com a expectativa que a sua adesão poderá levar ao sucesso e satisfação dos requisitos do cliente.

São tópicos abordados nesta sub-área: implementação do plano, contrato dos fornecedores da gestão, processo de medição, monitoramento e controle dos processos.

d. Revisão e avaliação

Essa atividade descreve as avaliações na busca da eficácia do processo global do projeto, contendo informações como: datas, pessoas envolvidas, ferramentas e métodos utilizados. Os objetivos principais são: determinar a satisfação dos requisitos, rever e avaliar o desempenho.

e. Fechamento

O projeto atinge seu fechamento, quando todos os planos e processos foram homologados e completados. A se estabelecer o fechamento é iniciada a execução das atividades de melhoria dos processos.

f. Evolução da Engenharia de Software

A importância da medição e seu papel nas melhores práticas de gestão são amplamente reconhecidos, e assim a sua importância só pode aumentar nos próximos anos. Medir a eficácia tornou-se um dos pilares da maturidade organizacional.

Um fator importante com relação a esta área é que, como pôde ser observada, ela possui atividades distribuídas durante todo o ciclo de vida do projeto. Isto significa que ela está ligada a praticamente todas as atividades que acontecem durante o projeto e, caso as atividades do gerenciamento estejam comprometidas, as atividades de qualquer fase do ciclo de vida podem ser afetadas reduzindo a qualidade de seus resultados finais.

Processo de Engenharia de Software

(Reis e Nunes, 2002) apud (Feiler, 1993) comenta que a Tecnologia de Processo de Software surgiu no final da década 80 e representou um importante passo em direção à melhoria da qualidade de software através de mecanismos que proporcionam o gerenciamento automatizado do desenvolvimento de software. Diversas teorias, conceitos, formalismos, metodologias e ferramentas surgiram nesse contexto, enfatizando a descrição de um modelo de processo de software que é automatizado por um ambiente integrado de desenvolvimento de software.

Comment [F266]: ?

A área de conhecimento do processo de Engenharia de Software está relacionada com a definição, implementação, controle, e proposta de mudança no próprio processo. Esta área pode ser visualizada em dois níveis. O primeiro nível engloba as atividades técnicas e gerenciais executadas durante a aquisição, desenvolvimento e manutenção do software. O segundo nível, tratado nesta área de conhecimento, considera as definições, implementações, gerenciamento e mudanças no próprio processo.

O primeiro nível é coberto por outro o KA's no Guia. O termo "processo de Engenharia de Software" pode ser interpretada de diferentes maneiras, e isso pode causar confusão

O processo de Engenharia de Software envolve vários outros processos, como o de desenvolvimento, o de gerenciamento, e o de qualidade. Esta área está ligada com qualquer parte do gerenciamento do processo de ciclo de vida do software, onde mudanças são propostas com o intuito de melhorar o produto ou até mesmo o processo de produção.

Ao contrário do que é suposto, essa área é importante não apenas para empresas grandes, mas também devem ser levadas em consideração para pequenas empresas, facilitando e auxiliando no crescimento dessas. O objetivo desse gerenciamento do processo é implementar novas práticas individuais, de projeto ou até mesmo organizacionais.

A área de processos de Engenharia de Software está dividida em quatro sub-áreas:

a. Mudança e Implementação do Processo

Esta atividade descreve a infra-estrutura, atividades, modelos, práticas e considerações sobre a implementação do processo e da mudança.

b. Definição do Processo

A definição do processo, conforme o guia, pode ser definido como uma política, um procedimento ou um padrão. Variáveis importantes a considerar incluem a natureza do trabalho, como por exemplo, a manutenção ou desenvolvimento. Importante salientar que o contexto do projeto e a organização irão determinar o tipo de processo que é mais útil.

c. Avaliação do Processo

Existem duas formas de avaliação para fazer as suposições sobre as ordens dos processos (contínuos ou escalonados), onde a organização define qual a mais pertinente para suas necessidades e objetivos. Tais processos são detalhados no capítulo X.

d. Medidas de Produtos e Processos

Embora a aplicação de medidas de Engenharia de Software possa ser complexa, especialmente em termos de modelagem e métodos de análise, existem vários aspectos de medição que são fundamentais por trás de muitas medidas avançadas e processos de análise. O guia traz como palavra chave para essa sub-área a norma ISO/IEC 15939 para descrever tais medidas e métodos para produtos e processos.

Uma das grandes dificuldades enfrentadas por esta área é que a implementação das suas práticas geralmente não traz benefícios em curto prazo, porém com a evolução dos processos a empresa vai aumentando o seu nível de maturidade e o desempenho das equipes e qualidade final dos produtos são beneficiados.

Métodos e ferramentas de engenharia

No SWEBOK, as ferramentas de desenvolvimento de software são ferramentas criadas para auxiliar no ciclo de vida do software. Essas ferramentas normalmente automatizam algumas atividades do processo de desenvolvimento, fazendo com que o analista concentre-se nas atividades que exigem maior trabalho intelectual (SWEBOK, 2004).

A sub-área Métodos de Engenharia de Software impõe uma estrutura sobre a atividade de desenvolvimento e manutenção de software com o objetivo de torná-la sistemática e mais propensa ao sucesso (SWEBOK, 2004).

O objetivo desta área de conhecimento é de pesquisar ferramentas e métodos que aumentem a produtividade dos desenvolvedores enquanto reduzem a ocorrência de falhas no desenvolvimento (FERNANDES, 2003). Embora existam manuais detalhados de ferramentas e inúmeros trabalhos de instrumentos inovadores, uma das dificuldades é a elevada taxa de mudança de ferramentas de software em geral (SWEBOK, 2004).

Esta KA cobre os processos do ciclo de vida do projeto de software e portanto relacionados a cada KA do guia, conforme ilustrado na figura 11.6.

Comment [F267]: Qual capítulo?

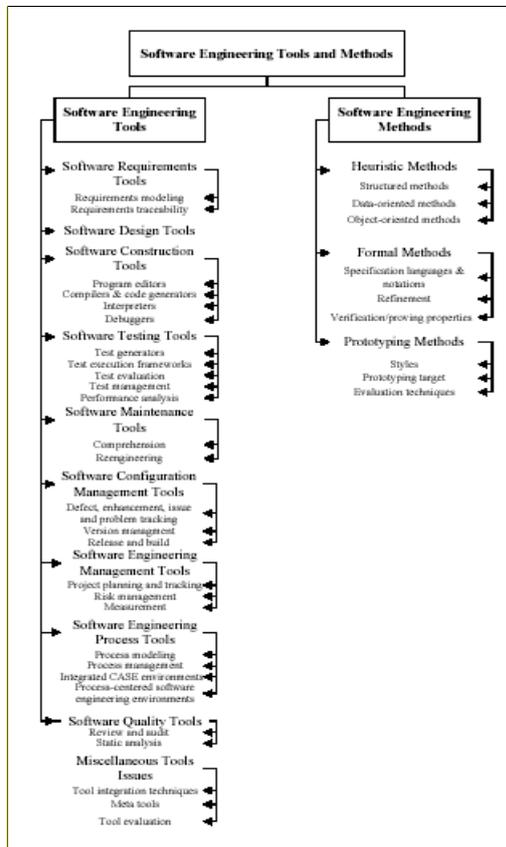


Figura 11.6. Distribuição dos tópicos relacionados aos Métodos e ferramentas de engenharia

Comment [F268]: Essa figura não está legível e ainda está em inglês. Outra coisa, essa figura existe para cada uma das áreas, porque você não a incluiu nas outras também?

a. Ferramentas de Engenharia de Software

Cada tópico ilustrado na figura 11.6 é referente em todos os KA's do guia, possuindo um tema adicional que aborda técnicas e ferramentas de integração, que são potencialmente aplicáveis a todas as classes de ferramentas.

b. Métodos de Engenharia de Software

É dividida em três temas:

- g) Heurísticos: abordagem informal;
- h) Matemáticos: abordagem formal;
- i) Protótipos: para a abordagem do software baseado em telas.

Formatted: Font: Not Bold

Formatted: Justified, Indent: Left: 0,54", Space Before: 6 pt, Bulleted + Level: 1 + Aligned at: 0,93" + Indent at: 1,18", Tab stops: 0,5", Left

Qualidade de software

A qualidade é relativa. O que é qualidade para uma pessoa pode ser falta de qualidade para outra. (Weinberg, 1994).

A ISO 9001 define a qualidade como "o grau em que um conjunto de características inerentes satisfaz as necessidades" (SWEBOK,2004).

Em relação à qualidade, o SWEBOK faz uma distinção entre técnicas estáticas e dinâmicas. As primeiras aparecem sob a área de conhecimento Qualidade, enquanto as últimas figuram na área de Testes. A norma

internacional ISO/IEC 25000 SQuaRE, que trata da qualidade de produtos de software, abrange esses dois tópicos (Koscianski, 2006).

Um dos principais objetivos da Engenharia de Software é melhorar a qualidade dos produtos de software, ela visa estabelecer métodos e tecnologias para construir produtos de software de qualidade dentro dos limites de tempo e recursos disponíveis. A qualidade de software está diretamente ligada com a qualidade do processo através do qual o software é desenvolvido, portanto, para se ter qualidade em um produto de software é necessário ter um processo de desenvolvimento bem definido, que deve ser documentado e acompanhado (SWEBOOK, 2004).

A avaliação da qualidade de produtos de software normalmente é feita através de modelos de avaliação de qualidade. Esses modelos descrevem e organizam as propriedades de qualidade do produto em avaliação. Os modelos de avaliação mais aceitos e usados no mercado são:

- CMMI (*Capability Maturity Model Integration*), proposto pelo CMM (*Capability Maturity Model*), tal modelo é detalhado no capítulo X.
- Norma ISO/IEC 9126, proposta pela ISO (*International Organization for Standardization*), tal modelo é detalhado no capítulo X.

As organizações desenvolvedoras desses modelos de qualidade fornecem selos de qualidade para as empresas que se submetem à avaliações e estiverem dentro dos padrões propostos. Esses selos são muito valorizados pelas empresas que compram software, e representam um diferencial competitivo no mercado. Porém, nem todas as empresas têm condições financeiras de bancar os custos de uma aquisição de um selo de qualidade, pois implantar um processo de qualidade em uma empresa envolve custos elevados. Contudo, é possível implantar boas práticas e desenvolver um processo de desenvolvimento organizado adaptando modelos de desenvolvimento conhecidos, despendendo menos recursos e provendo um mínimo de sistematização no desenvolvimento de software, a fim de se ter maior qualidade.

A área de Qualidade de Software, segundo o SWEBOOK, é dividida em três tópicos que serão rapidamente descritos a seguir:

a. Fundamentos da Qualidade de Software

Este tópico abrange a definição de qualidade, buscando acordar os requisitos da qualidade, bem como efetuar uma comunicação clara como o engenheiro de software sobre tais requisitos.

Os aspectos éticos do trabalho com software têm se tornado mais evidentes com nossa dependência da tecnologia; toda uma nova classe de problemas surgiu com os crimes de computador. Respostas sociais, éticas e de legislação estão sendo desenvolvidas para procurar tratar adequadamente cada caso (KOSCIANSKI, 2006). Este tópico tem com subáreas: Engenharia de Software Cultura e Ética, valor e custo da qualidade, modelos e características da qualidade e a melhoria da qualidade.

b. Processo de Gerenciamento da Qualidade de Software

Tal tópico abrange todos os aspectos de construção do produto. São abordados todos os elementos de um projeto, como: ferramentas para controle de versão e linguagens, metodologias para revisão do produto, técnicas organizacionais e de administração de pessoas etc.

O propósito da subárea é assegurar que os objetivos planejados no início do projeto sejam cumpridos, se aplicando a todas as perspectivas do processo de software. Definindo processos, responsáveis, produtos e recursos.

Alguns dos processos desta subárea são definidos pelo padrão IEEE 12207, sendo eles: garantia de qualidade, verificação, validação, revisão e auditoria.

c. Considerações Práticas sobre a Qualidade de Software

Neste tópico são apresentadas as recomendações gerais sobre como transcorre a execução das atividades relacionadas com qualidade.

Na estrutura deste tópico estão as subáreas:

- 2 Requisito de qualidade de software: menciona itens como “fatores de influência” sobre requisitos, segurança do funcionamento e as conseqüências que as falhas podem causar;
- 3 Caracterização de defeitos: verifica a não-conformidade aos requisitos;
- 4 Técnicas de gestão de qualidade de software: podem ser orientadas a pessoas (revisões e auditorias), estáticas (não envolvem execução do produto), dinâmicas (efetuados durante a execução do produto) e técnicas analíticas (utilização de métodos formais).

Formatted: Justified, Indent: Left: 0,54", Space Before: 6 pt, Bulleted + Level: 1 + Aligned at: 0,74" + Indent at: 0,99", Tab stops: 0,5", Left

Comment [F269]: Qual capítulo?

Comment [F270]: Qual capítulo?

Comment [F271]: Esse parágrafo ficou meio fora de contexto

Formatted: Indent: Left: 0,59", No bullets or numbering

Formatted: Justified, Indent: Left: 0,59", First line: 0", Space Before: 6 pt, Bulleted + Level: 1 + Aligned at: 0,74" + Indent at: 0,99", Tab stops: 0,5", Left

5. Medição da qualidade: inclusão de medidas para determinar o grau de qualidade atingido pelo produto. Como por exemplo, o proposto pela norma SQuaRE, onde os valores desejados para as medidas sejam estabelecidos no início do projeto, ao se definir os requisitos.

Disciplinas relacionadas

Para delimitar a engenharia de software se faz necessário identificar as disciplinas com que ela compartilha uma fronteira comum. Nesta seção é feita a identificação em ordem alfabética das disciplinas relacionadas, conforme ilustrada na figura 11.7.



Figura 11.7. Disciplinas relacionadas com a Engenharia de Software

Formatted: Font: Not Bold

a. Engenharia da Computação

Conforme relatório do Computing Curricula 2001 Computer Science a engenharia da computação incorpora a tecnologia e ciências de concepção, construção, implementação e manutenção de componentes de software e hardware dos sistemas de computação e controlados por computador. Destacam-se como áreas de conhecimento: algoritmos, arquitetura e organização de computadores, engenharia de sistemas de computadores, entre outras.

b. Ciências da Computação

O relatório final Computing Curricula 2001 Computer Science identifica diversas áreas de conhecimento que estão relacionadas com a ciência da computação, tais como: sistemas operacionais, linguagem de programação, computação gráfica, engenharia de software, entre outras.

c. Gerenciamento

Segue recomendações para MBA definidos pelo Conselho Europeu, que inclui como áreas de conhecimento relacionadas: contabilidade, finanças, marketing, direito, gestão de recursos humanos, entre outras.

d. Matemática

É recomendado para o engenheiro de software, conforme o guia, os conhecimentos de álgebra linear, equações diferenciais, probabilidade, estatística, entre outras.

e. Gestão de Projetos

Segue as recomendações conforme o guia PMBOK Guide 2006, sendo o mesmo detalhado no capítulo X deste livro. São conhecimentos relacionados pelo guia: gestão de custos, de riscos, de qualidade, entre outros.

Comment [F272]: Qual capítulo?

f. Gestão de Qualidade

A gestão da qualidade é definida na norma 9000. O guia recomenda o conhecimento para a ISO 9000, 9001, 9004.

g. Ergonomia

É uma disciplina científica relacionada com a compreensão das interações entre humanos e outros elementos de um sistema. Tende como conhecimentos relacionados: computação gráfica, processo de desenvolvimento, técnicas de aplicação, entre outros.

h. Engenharia de Sistemas

O Conselho Internacional de Engenharia de Sistemas (INCOSE) afirma que “ Engenharia de Sistemas é uma abordagem interdisciplinar que permitem a realização de sistemas bem sucedidos”. São conhecimentos relacionados: verificação das necessidades dos clientes, funcionalidade necessárias no início do ciclo de desenvolvimento, documentação de requisitos, entre outros.

11.2.2. SWEBOK 2010

Novos estudos estão sendo realizados para a atualização do Guia de Conhecimento da Engenharia de Software - *Guide to the Software Engineering Body of Knowledge* (SWEBOK) com a intenção de incluir novas área de conhecimento e rever outras.

O objetivo principal da atual revisão do Guia SWEBOK é a adição de um KA sobre as práticas profissionais, um assunto atualmente abrangida pela certificação CSDP, além de acrescentar KA's sobre assuntos relacionados com os engenheiros de software e educação para graduação. Além de:

- k) Remoção de três disciplinas relacionadas: Ciência da Computação, Matemática e Ergonomia;
- l) Adicionado material sobre Interfaces Humano-Computador no design de software e Teste de Software;
- m) Remoção da seção Ferramentas e métodos de Engenharia de Software (distribuídos para outras áreas de conhecimento);
- n) ***** Redistribuição de matérias entre as áreas de conhecimento;
A atualização completa do SWEBOK Guide tem previsão para o primeiro semestre de 2010.

11.3. Tópicos de pesquisa

Algumas instituições destacam-se pela aplicação do SWEBOK para a formação de graduados em Engenharia de Software, como por exemplo: Universidade de New South Wales (Austrália), Universidade McMaster (Canadá), o Instituto de Tecnologia de Rochester (E.U.) e a Universidade de Sheffield (Reino Unido).

Dentro das aplicações do SWEBOK por instituições pelo mundo, podemos citar:

- g) Canadian Information Processing Society: publicação dos critérios de aceitação da Engenharia de Software por programas universitários de graduação na área.
- h) A norma ISO 9000 (Gestão da Qualidade) aplicados à Engenharia de Software (ISO/IEC90003).
- i) Efetivação de licenças de engenheiros de software baseados no guia: Conselho de Engenheiros Profissionais do Texas, Associação de Engenheiros Profissionais e geocientistas da Columbia Britânica (APEGBC).
- j) ***** A ACM e IEEE desenvolveram conjuntamente e aprovaram um Código de Ética e Exercício Profissional da Engenharia de Software.

11.4. Sugestão de leitura

Para saber um pouco mais sobre os estudos conduzidos pela IEEE que colabora **com o** no incremento da prosperidade mundial, promovendo a engenharia de criação, desenvolvimento, integração, compartilhamento e o conhecimento aplicado no que se refere à ciência e tecnologias da eletricidade e da informação, em benefício da humanidade e da profissão, consulte: <http://www.computer.org/portal/web/guest/home>.

Com interesse em consultar o guia completo do SWEBOK, suas certificações e suas áreas de conhecimento, consulte: <http://www.swebok.org/index.html>, onde poderá baixar a versão digital do guia 2004.

O livro de Koscianski e Soares (Qualidade de Software, Novatec, 2006) oferece uma cobertura detalhada das métricas da qualidade, metodologias e técnicas mais modernas para o desenvolvimento de software. O livro aborda a definição de qualidade de software, normas e organismos normativos, métricas e conteúdo que se refere principalmente aos processos de desenvolvimento de software.

Formatted: Justified, Indent: Left: 0,48", Hanging: 0,01", Space Before: 6 pt, Bulleted + Level: 1 + Aligned at: 0,74" + Indent at: 0,99", Tab stops: 0,5", Left

Comment [F273]: Essa fonte e o espaçamento estão corretos?

Formatted: Justified, Indent: Left: 0,58", Hanging: 0,01", Space Before: 6 pt, Bulleted + Level: 1 + Aligned at: 0,74" + Indent at: 0,99", Tab stops: 0,5", Left

Comment [F274]: Essa fonte e o espaçamento estão corretos?

Comment [F275]: Está formatado diferente do link abaixo.

Uma excelente fonte de informação sobre testes de software, comentado na seção 11.2.2.4, pode ser consultada no livro de Rios e Moreira (Teste de Software, segunda edição, Alta Books, 2006), para aqueles que estão se iniciando no assunto de teste de software.

11.5. Exercícios

Comment [F276]: Essa fonte e o espaçamento estão corretos?

1. O que é o Guide to the SWEBOK e por qual entidade é produzido?
2. Quais são os cinco principais objetivos do SWEBOK?
3. Quantas e quais são as áreas de conhecimento em que o SWEBOK é dividido?
4. Como o objetivo “Acesso por Tópicos às Referências” é tratado em cada Área de Conhecimento?
5. O que é um Requisito de Software?
6. De que trata a área de conhecimento Requisitos de Software? Qual a sua relação com o problema que deve ser resolvido pelo software?
7. Qual a importância de uma boa Especificação de Requisitos para a qualidade do software?
8. De que trata a área de conhecimento Design de Software? Como distingui-la do que comumente chamamos de Projeto de Software?
9. De que trata a área de conhecimento Construção de Software?
10. De que trata a área de conhecimento Teste de Software? Todos os tipos de teste de software estão exclusivamente tratados nesta área de conhecimento?
11. De que trata a área de conhecimento Manutenção de Software?
12. Todas as manutenções de software referem-se aos erros ocorridos no software? Justifique sua resposta.
13. Como são classificadas as manutenções, de acordo com o SWEBOK? Em sua opinião, quais as categorias de manutenção devem ocorrer com maior frequência, idealmente?
14. Descreva o ciclo de atividades dentro do processo de manutenção, segundo o SWEBOK.
15. De que trata a área de conhecimento Gerenciamento de Configuração de Software?
16. Quais as atividades do processo de Gerenciamento de Configurações, segundo o SWEBOK?
17. De que trata a área de conhecimento Gerenciamento do Processo de Software?
18. De que trata a área de conhecimento Processo de Software?
19. De que trata a área de conhecimento Ferramentas e Métodos?

Comment [F277]: Existe distinção mesmo?

Referências

Koscianski, A. and Soares, M.S. (2006). “Qualidade de Software” - 2ª edição, Novatec. São Paulo - SP

Comment [F278]: Fonte, numeração e espaçamento corretos?

Naur, P. and Randell, B. E. (1968). “Software Engineering: Report on a Conference Sponsored by the NATO Science Committee. Technical report”, NATO, Garmisch, Germany.

Comment [F279]: Uniformizar a formatação das referências.

Pressman, R. S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

Pressman, R. S. **Engenharia de Software**. 5. ed. Rio de Janeiro: McGraw-Hill, 2002.

SWEBOK. Guide to the Software Engineering Body of Knowledge. 2004 Version. project of the IEEE Computer Society Professional Practices Committee. Disponível em: <<http://www.swebok.org/>>. Acesso em: 22 Ago. 2009.

Tavares, A. L. O. and Eckel, A. P. and Scarpa, C. and Vedrame R. "Engenharia de Software: Uma Visão Geral". Curso de especialização em Engenharia de Software de Projetos de Software - Universidade do Sul de Santa Catarina (UNISUL) – Palhoça, SC – Brasil

Weinberg, G.M. *Software com Qualidade*. Makroon Books, 1994.

Rios, E; Moreira, T. Teste de Software, Segunda edição, Alta Books, 2006.

Feiler, P.H; Humphrey, W.S. Software Process Development and Enactment: Concepts and Definitions. In: II International Conference on the Software Process, 1993. Berlin.

Reis, R.Q; Reis, C.A; Nunes, D. J. Automação no Gerenciamento do processo de Engenharia de Software. Departamento de Informática, Universidade Federal do Pará (UFPA), Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), Belém (PA), outubro de 2002.

Ribeiro, D.A. Escolha de uma das áreas de Engenharia de Software do SWEBOOK Centro de Informática – Universidade Federal de Pernambuco (UFPE), Recife – PE – Brasil

Capítulo

12

Gerenciando Projetos de Software

Paula Geralda Barbosa Coelho Torreão, MSc, PMP¹⁵

"De todos os monstros que fazem parte dos pesadelos do nosso folclore, nenhum é mais terrível do que o lobisomem, porque ele se transforma inesperadamente de algo familiar num horror. Projetos de software têm algo deste caráter: geralmente são claros e inocentes, mas são capazes de transformar-se em um mostro de prazos perdidos, orçamentos ultrapassados e produtos cheio de defeitos." (Frederick P. Brooks)

Projetos de software ainda hoje têm tido baixos índices de sucesso. As causas de fracasso dos projetos são diversas e conhecidas desde o surgimento do computador, mas muitas delas continuam ocorrendo. O relatório Chaos Report do Standish Group do ano de 2004, ao analisar os projetos de TI que falharam, apresentou que para a maioria deles, a principal causa não foi a falta de recursos financeiros ou acesso à tecnologia, mas sim, a falta de conhecimento em gestão de projetos¹⁶. Este cenário é preocupante quando o mesmo relatório de 2009 aponta que somente 32% dos projetos têm sucesso e que este percentual reduziu 3% em relação ao ano de 2006 [Standish Group, 2004, 2006, 2009].

Entretanto, a aplicação das melhores práticas de Gerenciamento de projetos nas organizações vem contribuindo para a mudança deste cenário e tem impactado positivamente no resultado dos projetos. O sucesso de um projeto de software é influenciado pela forma como ele é gerenciado e cada vez mais as organizações de TI e de outras áreas consideram o Gerenciamento de projetos essencial para o sucesso organizacional.

¹⁵ PMP® (Project Management Professional) é uma marca registrada do Project Management Institute.

¹⁶ Neste texto o termo gestão de projetos é igual ao termo Gerenciamento de projetos.

Este capítulo apresenta um entendimento sobre Gerenciamento de projetos, descrevendo suas definições básicas, evolução e relevância; e mostra a visão do Project Management Institute (PMI¹⁷) sobre Gerenciamento de projetos.

14.1. Definições básicas

Projeto é um instrumento fundamental para qualquer atividade de mudança e geração de produtos e serviços. Ele pode envolver uma ou mais pessoas e ter a duração de horas, dias ou anos. Um projeto é um empreendimento único, com início e fim definidos, que utiliza recursos limitados e é conduzido por pessoas, visando atingir metas e objetivos pré-definidos. Ele inclui identificação dos requisitos, adaptação ao longo do projeto às expectativas e necessidades das partes interessadas (*stakeholders*) e equilíbrio de restrições conflitantes do projeto tais como escopo, qualidade, cronograma, orçamento, recursos e riscos. A relação entre estas restrições ocorre no projeto de tal forma que se alguma delas mudar, pelo menos uma outra será afetada impactando os resultados do projeto [PMI 2008, Torreão 2005].

O projeto pode ser definido por características distintas como temporário, único e progressivo. A característica de ser temporário é muito importante, pois todo projeto tem início e fim definidos. O projeto termina quando os objetivos para o qual foi criado são atingidos ou quando se torna claro que os objetivos do projeto não serão ou não poderão mais ser atingidos ou a necessidade do projeto não existe mais. Ser único significa que cada projeto cria um produto, serviço ou resultado exclusivo. Os projetos envolvem características singulares, jamais realizadas anteriormente, como por exemplo, diferentes materiais, fornecedores, equipe, e por isto é único. Um projeto é progressivo porque à medida que é mais bem compreendido, ele é progressivamente elaborado, ou seja, maior é o detalhamento das características peculiares que o distinguem como único [Dinsmore e Cavalieri 2003; PMI 2008].

Um projeto para ser executado precisa ser gerenciado. Segundo Koontz e O'Donnel [1980], gerenciar consiste em executar atividades e tarefas que têm como propósito planejar e controlar atividades de outras pessoas para atingir objetivos que não podem ser alcançados caso as pessoas atuem por conta própria, sem o esforço sincronizado dos subordinados.

Segundo o PMI [2008], o Gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos. Para facilitar o gerenciamento do projeto ele deve ser dividido em fases que constituem seu ciclo de vida [Dinsmore e Cavalieri 2003].

O ciclo de vida do projeto serve para definir o início e o fim do projeto e definem qual o trabalho (atividades) deve ser realizado em cada fase (ou etapa) e quem deve estar envolvido. Ele descreve o conjunto de processos que devem ser seguidos para que o projeto seja bem gerenciado [Dinsmore e Cavalieri 2003; PMI 2008].

O sucesso de um projeto é medido pela qualidade do produto e do projeto, pontualidade, conformidade orçamentária e grau de satisfação do cliente [PMI 2008]. O gerente de projeto é responsável pelo gerenciamento do projeto e conseqüentemente pelo seu sucesso ou fracasso, no entanto, vale ressaltar, que o sucesso do projeto depende também do comprometimento de toda a equipe do projeto.

O gerente deve ser designado desde o início do projeto e deve ter o apoio visível da alta administração para atingir os objetivos do projeto. Ele deve ter a sua competência reconhecida pelos demais interessados no projeto, embora não precise ter profundo conhecimento técnico uma vez que sua competência está mais voltada para o entendimento geral e não para o específico [Dinsmore e Cavalieri 2003; PMI 2008].

Segundo o PMI [2008], um gerente de projeto deverá estar atento a todo o contexto que dirá respeito à sua gestão, ao ciclo de vida do projeto, aos *stakeholders* (as partes interessadas, envolvidos direta e indiretamente com o projeto), aos seus detalhes e perspectiva global, às influências organizacionais e às influências ambientais-sócio-econômicas. Destacam-se como habilidades gerenciais inerentes ao gerente de projetos: a liderança forte, a comunicação, a negociação, a resolução de problemas, flexibilidade, bom senso e a influência na organização. Além das habilidades da área específica e das proficiências ou competências de

¹⁷ PMI® é uma marca registrada do Project Management Institute.

gerenciamento geral, o gerente de projeto para ter um gerenciamento eficaz precisa ter as seguintes características:

- Conhecimento - saber sobre Gerenciamento de projetos;
- Desempenho – ser capaz de realizar enquanto aplica o seu conhecimento em Gerenciamento de projetos;
- Pessoal – ter comportamento efetivo na execução do projeto ou de alguma atividade relacionada ao projeto, que abrange: atitudes, personalidade, liderança; capacidade de orientar a equipe ao mesmo tempo que atinge os objetivos e equilibra as restrições do projeto.
O gerente do projeto possui várias atividades e responsabilidades, como por exemplo:
- desenvolver o plano de gerenciamento do projeto e todos os planos componentes relacionados;
- manter o projeto na direção correta, dentro do planejado, em relação ao cronograma e orçamento;
- definir e controlar os objetivos do projeto;
- definir e controlar os requisitos do produto;
- identificar, analisar e monitorar os stakeholders (equipe, clientes, usuários, patrocinadores, outros gerentes, fornecedores, qualquer parte interessada ou afetada pelo projeto);
- definir prioridades; coordenar interações entre os stakeholders do projeto;
- fazer a comunicação efetiva do projeto;
- identificar, controlar e responder aos riscos do projeto;
- definir e avaliar os fatores críticos de sucesso do projeto, seus pontos fortes e fracos;
- alocar e gerenciar pessoas e recursos (orçamento, materiais);
- verificar o esforço, avaliar o projeto e a equipe com métricas;
- assegurar que os produtos do projeto atendam aos critérios de qualidade e que estejam de acordo com os padrões estabelecidos;
- formalizar a aceitação dos artefatos resultantes de cada fase do ciclo de vida do projeto;
- elaborar relatórios de avaliação e de acompanhamento da situação do projeto;
- participar de reuniões de acompanhamento e de revisão do projeto;
- realizar reuniões de lições aprendidas.

O gerente de projetos cada vez mais ganha destaque dentro das organizações pela evolução e relevância do Gerenciamento de projetos. A profissão de Gerenciamento de projetos é emergente e bastante promissora, principalmente dentro das organizações de TI [Martins 2003; PMI 2009].

14.2. Evolução do Gerenciamento de projetos

Projetos vêm sendo realizados desde os primórdios da civilização. A construção das Pirâmides do Egito, depois de 2780 a.C. [Vicentino 1997], por exemplo, foi um grande projeto. A exploração da lua, a construção do primeiro computador, da Estátua da Liberdade, da Torre Eiffel, da Muralha da China, do Canal do Panamá, da cidade de Brasília e da Bomba Atômica, são exemplos históricos de grandes projetos.

Projetos têm sido planejados e

Formatted: Font: 14 pt, Bold, Kern at 14 pt

Formatted: Font: 14 pt, Bold, Kern at 14 pt

executados pelas organizações para criar novos produtos/serviços e introduzir mudanças e inovações em seus processos. No entanto, para que um projeto seja realizado de forma eficaz, é necessária a organização do trabalho demandado [Martins 2003].

Voltando no tempo temos, na última metade do século XIX, um crescente aumento na complexidade dos novos negócios em escala mundial surgindo assim os princípios da gerência de projetos¹⁸. A Revolução Industrial alterou profundamente a estrutura econômica do mundo ocidental e teve como uma das suas principais conseqüências o desenvolvimento do capitalismo industrial. As relações de produção foram drasticamente modificadas e iniciou-se assim, uma cadeia de transformações, que tornou cada vez mais exigente a tarefa de gerir as novas organizações econômicas [Sisk 1998].

Conseqüentemente, a partir daí surgiu uma grande necessidade de sistematizar e orientar a forma de gerir estas organizações [Martins 2003]. Os projetos, em grande escala do governo, eram o ímpeto para tomar as decisões importantes que se transformaram em decisões de gerenciamento [Sisk 1998].

Nos EUA, a primeira grande organização a praticar tais conceitos foi a *Central Pacific Railroad*¹⁹, que começou suas atividades no início da década de 1870, com a construção da estrada de ferro transcontinental. De repente, os líderes do negócio se depararam com a perigosa tarefa de organizar as atividades de milhares de trabalhadores, a manufatura e a montagem de quantidades não previstas de matéria-prima [Sisk 1998].

Frederick Taylor (1856-1915), no início do século XX, iniciou seus estudos de forma detalhada sobre trabalho. Ele aplicou raciocínio científico para mostrar que o trabalho pode ser analisado e melhorado focando em suas partes elementares. Ele aplicou sua teoria às atividades encontradas na indústria de aço (por exemplo, carregar areia, levantar areia) [Sisk 1998].

Antes de Taylor, a única maneira de melhorar a produtividade era exigir dos trabalhadores mais horas de dedicação ao trabalho. Taylor ocupa um lugar importante na história da gerência de projetos e de acordo com a escritura em seu túmulo ele é "o pai do gerenciamento científico" [Sisk 1998].

O sócio de Taylor, Henry Gantt (1861-1919), estudou detalhadamente a ordem de operações no trabalho. Seus estudos de gerenciamento focaram na construção de um navio durante a II Guerra Mundial. Gantt construiu diagramas com barras de tarefas e marcos, que esboçam a seqüência e a duração de todas as tarefas em um processo [Sisk 1998].

Os diagramas de Gantt provaram ser uma ferramenta analítica tão poderosa para gerentes que se mantiveram virtualmente inalterados por quase cem anos. Não foi realizada alteração até antes dos anos 90, onde linhas de ligação foram adicionadas às barras de tarefa que descrevem dependências mais precisas entre as tarefas. Taylor e Gantt, e outros estudiosos ajudaram a desenvolver o processo de gerência como uma função distinta de negócio que requer estudo e disciplina [Sisk 1998].

Nas décadas seguintes à II Guerra Mundial, as estratégias de marketing, a psicologia industrial, e as relações humanas começaram a ser partes integrantes do gerenciamento do negócio, da administração das empresas. Desta forma, a complexidade dos projetos demandou novas estruturas organizacionais. Complexos Diagramas de Rede, chamados de Gráficos de PERT (*Program Evaluation and Review Technique*) e o método de Caminho Crítico (*Critical Path Method* - CPM) foram introduzidos, oferecendo aos gerentes maior controle sobre os projetos. Rapidamente, essas técnicas foram difundidas entre gerentes que procuravam novas estratégias e ferramentas de gerenciamento, que permitissem o desenvolvimento de projetos em um mundo competitivo e de mudanças rápidas [Sisk 1998].

Em pouco tempo, essas técnicas espalharam-se para todos os tipos de indústria. Logo, líderes de projeto procuraram novas estratégias e ferramentas para gerenciar seu crescimento e o dinamismo das mudanças em um mundo competitivo. As teorias gerais do sistema da ciência então começaram a serem aplicadas às interações do negócio [Sisk 1998].

Os negócios começaram a serem vistos como um organismo humano, com esqueleto, sistema muscular, circulatório, nervoso e por aí em diante. Esta visão de organismo humano implica que para um negócio sobreviver e prosperar todas as suas partes funcionais precisam trabalhar juntas visando metas específicas, ou projetos [Sisk 1998].

No início dos anos 60, o Gerenciamento de projetos foi formalizado como ciência

18 Neste texto o termo gerência de projetos é igual ao termo Gerenciamento de projetos.

19 Central Pacific Railroad na Web: cpr.org.

[Prado 2000]. Os negócios e outras organizações começaram a enxergar o benefício do trabalho organizado em torno dos projetos e a entender a necessidade crítica para comunicar e integrar o trabalho através de múltiplos departamentos e profissões [Sisk 1998].

Em 1969, no auge dos projetos espaciais da NASA, um grupo de cinco profissionais de gestão de projetos, da Filadélfia, Pensilvânia, nos EUA, se reuniu para discutir as melhores práticas e Jim Snyder fundou, nos EUA, o Project Management Institute (PMI). O PMI é a maior instituição internacional dedicada à disseminação do conhecimento e ao aprimoramento das atividades de gestão profissional de projetos atualmente [PMI 2009, Sisk 1998].

Nas décadas seguintes, o Gerenciamento de projetos, começou a tomar sua forma moderna. Enquanto vários modelos de negócio desenvolveram-se neste período, todos eles compartilharam uma estrutura de suporte comum: projetos são liderados por um gerente de projetos, que põe pessoas juntas em um time e assegura a integração e comunicação de fluxos de trabalho através de diferentes departamentos [Sisk 1998].

Segundo Prado [2000], a boa prática de gerenciamento de projetos produz resultados expressivos para as organizações como: (1) redução no custo e prazo de desenvolvimento de novos produtos; (2) aumento no tempo de vida dos novos produtos; (3) aumento de vendas e receita; (4) aumento do número de clientes e de sua satisfação e (5) aumento da chance de sucesso.

As organizações bem sucedidas reconhecem o valor do Gerenciamento de Projetos porque colhem bons resultados através da aplicação das boas práticas de gestão em seus projetos. Elas sabem que gerenciar projetos com eficiência constitui-se não apenas um grande desafio dos dias atuais, mas é fator crítico para o sucesso, sobrevivência e crescimento futuro das organizações.

14.3. Relevância do Gerenciamento de projetos

Formatted: Font: 14 pt, Bold, Kern at 14 pt

Formatted: Font: 14 pt, Bold, Kern at 14 pt

Atualmente, há uma grande demanda por Gerenciamento de Projetos. O PMI estima que aproximadamente 1/3 do PIB mundial será gasto em projetos de 2009 a 2013, e que existem mais de 20 milhões de praticantes de Gerenciamento de Projetos pelo mundo [PMI 2009]. Estes dados refletem a necessidade da gestão de projetos de forma organizada e inovadora pelas organizações.

Na área de TI este assunto assume a cada dia uma importância maior. Isto se deve, em parte, pelo entendimento de que parte significativa do insucesso em projetos de software está relacionada com uma má gerência de projetos ou, algumas vezes, por uma ausência completa de gerenciamento [Johnson 2001a]. O Relatório CHAOS [Johnson 2001; Standish Group, 2004, 2006, 2009], tradicional estudo sobre sucessos e fracassos em projetos de TI realizado pelo *Standish Group International*²⁰, em 2001 já apontava que os três primeiros fatores que contribuíram para o sucesso do projeto de software foram: suporte executivo, envolvimento do usuário e experiência do gerente de projetos. Juntos, eles contabilizam 48% de chance do projeto ter sucesso. Este relatório cita que 97% dos projetos de sucesso têm um gerente experiente na liderança. Em 2003, o relatório CHAOS, analisou 13.552 projetos norte-americanos e mostrou que a taxa de sucesso atingiu 34%, mais que o dobro dos 16% obtidos em 1994, primeiro ano da pesquisa [Johnson 2001]. Os fracassos ficaram em 15%, cerca de metade dos 31% do primeiro estudo. A perda de dólares entre os projetos em 2002 foi estimada em 55 bilhões de dólares, frente aos 255 bilhões de dólares investidos em projetos em 1994, o desperdício foi de 140 bilhões de dólares (80 bilhões em projetos fracassados). O atraso, aspecto importante da falta de gerenciamento dos prazos, segundo o *Standish Group*, aumentou para 82%, em 2000 era de 63%. No relatório de 2004, a principal causa dos fracassos em projetos não foi a falta de recursos financeiros ou acesso à tecnologia, mas sim, a falta de conhecimento em gestão de projetos.

Ainda se tem muito a ser feito na aplicação efetiva do Gerenciamento de Projetos de Software, pelo menos é o que comprova o relatório de 2009. Embora haja algumas controvérsias em relação a opiniões de profissionais da área de TI com os dados apresentados neste relatório, ele aponta que somente 32% dos projetos têm sucesso (este percentual reduziu 3% em relação ao ano de 2006), 24% falham e 44% são mudados. Destes

²⁰ Standish Group na Web: www.standishgroup.com/

45% ultrapassam o orçamento e 63% não terminam no prazo. Este relatório aponta ainda que, na média, somente 67% das funcionalidades do software são entregues. Diante destes resultados, é notória a importância do gerenciamento de projetos e a sua utilização de forma profissional. Construir um bom software requer bom gerenciamento do projeto de software.

14.4. Conhecendo o PMI

O PMI é uma associação sem fins lucrativos, cujo principal objetivo é difundir a gestão de projetos no mundo, de forma a promover ética e profissionalismo no exercício desta atividade, visando promover e ampliar o conhecimento existente sobre gerenciamento de projetos, assim como melhorar o desempenho dos profissionais e organizações nesta área [Martins 2003]. Esta associação ocupa uma posição de liderança global no desenvolvimento de padrões para a prática da profissão de gerenciamento de projetos em todo o mundo.

.....continua

14.5. Gerenciamento de Projetos na visão do PMI

O gerenciamento de projetos, na visão do PMI, de acordo com o PMBOK Guide, 4ª edição [PMI 2008], identifica e descreve as principais áreas de conhecimento e práticas. Cada uma destas áreas (no total de 9) é descrita através de processos (no total de 42), e se refere a um aspecto a ser considerado dentro da gerência de projetos. As áreas de conhecimento de gerenciamento são: Gerenciamento de Integração do Projeto, Gerenciamento de Escopo do Projeto, Gerenciamento do Tempo do Projeto, Gerenciamento do Custo do Projeto, Gerenciamento da Qualidade do Projeto, Gerenciamento de Recursos Humanos do Projeto, Gerenciamento de Comunicação do Projeto, Gerenciamento do Risco do Projeto e Gerenciamento de Contratação do Projeto.

...continua

14.6. Tópicos de Pesquisa

14.7. Sugestões de Leitura

14.8. Exercícios

Referências

- Dinsmore, C. e Cavalieri, A. (2003). Como se Tornar um Profissional em Gerenciamento de Projetos: Livro-Base de “Preparação para Certificação PMP_ - Project Management Professional”. Rio de Janeiro. QualityMark.
- Gates, Willian H.; (1999). III, Business @ the Speed of Thought, New York, NY: Warner Books.
- Johnson, J. (2001). Micro Projects Cause Constant Change, The Standish Group International, Inc. Disponível em: <<http://www.xp2001.org/xp2001/conference/papers/Chapter30-Johnson.pdf>>. Acesso em 01 de nov. 2001.
- Kerzner, H. (2001). Project Management – A Systems Approach to Planning, Scheduling, and Controlling. New York: John Wiley & Sons.
- Koontz, H. e O’Donnel, C. (1980). Os Princípios de Administração: Uma Análise das Funções Administrativas. São Paulo, Pioneira.

Formatted: Font: 14 pt, Bold, Kern at 14 pt

Formatted: Font: 14 pt, Bold, Kern at 14 pt

Formatted: Font: 14 pt, Bold, Kern at 14 pt

Formatted: Font: 14 pt, Bold, Kern at 14 pt

Formatted: Font: (Default) Times-Roman, 11,5 pt, Portuguese (Brazil)

- Martins, L.; (2003) Gestão Profissional de Projetos. Disponível em: <http://www.ietec.com.br/ietec/techoje/techoje/gestaodeprojetos/2003/10/10/2003_10_10_0003.2xt/-template_interna>. Acesso em 01 de abr. 2004.
- Melhoramentos, de 2008.
- Nagel, B. (2003). 10 Hottest Certifications for 2004, CertCities.com, Dezembro, 2003. Disponível em: <<http://certcities.com/editorial/features/story.asp?EditorialsID=76>>. Acesso em 01 de fev. 2004.
- Neto, J. e Bocoli, F. (2003). SUCESSOSW = CMM2 + PMBOK. PMI Journal, Publicação da Seção do PMI-RS. Número 5, Maio 2003. pág: 2-11. Disponível em: <http://www.pmir.org/PMI20_Frame.htm>. Acessado em 01 de fev. 2004.
- PMI - Project Management Institute (2008). Um guia do conhecimento em gerenciamento de projetos (PMBOK). 4ª edição. Project Management Institute, Inc, 2008. Disponível em: <<http://www.pmi.org>>. Acessado em 20 de nov. 2009.
- PMI - Project Management Institute (2009). Site oficial do PMI. Disponível em: <<http://www.pmi.org>>. Acessado em 20 de nov. 2009.
- Prado, D. (2000). Gerenciamento de projetos nas Organizações, Vol-I, Belo Horizonte, FDG.
- Sandeep, M.; (2002). The Accidental Profession Comes of Age. Disponível em: <<http://www.standards.org.au/STANDARDS/NEWSROOM/TAS/2002-06/PROJECT/PROJECT.HTM>>. Acessado em 01 de abr. 2003.
- Sisk, T. (1998). History of Project Management. Disponível em: <<http://office.microsoft.com/downloads/9798/projhistory.aspx>>. Acessado em 01 de abr. 2003.
- Standish Group (2004). The CHAOS Report [S.I.]. Disponível em: <http://www.standish-group.com>. Acessado em 02 de dez. 2004.
- Standish Group (2006). The CHAOS Report [S.I.]. Disponível em: <http://www.standish-group.com>. Acessado em 20 de dez. 2009.
- Standish Group (2009). The CHAOS Report [S.I.]. Disponível em: <http://www.standish-group.com>. Acessado em 20 de nov. 2009.
- Termini, M. (2003). Gerentes de Projetos Ganham espaço com a crise Mundial. Entrevista de Stela Campos, Valor Econômico - 29.4.2003. Reportagem com professor Michael Termini, da Universidade de Richmond. PMI Journal, Publicação da Seção do Rio Grande do Sul, Brasil - PMI-RS Número 5, Maio 2003. pág: 32-34. Disponível em: <http://www.pmir.org/PMI20_Frame.htm>. Acessado em: 01 de abr. 2003.
- Torreão, P. (2005). Project Management Knowledge Learning Environment: Ambiente Inteligente de Aprendizado para Educação em Gerenciamento de Projetos. Recife, 2005. 146p. Dissertação de Mestrado do Curso de Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco.
- Vicentino, C. (1997). História Geral. São Paulo, Editora Scipione.
- Vieira, E. (2002). Gerenciando Projetos na Era de Grandes Mudanças – Uma breve abordagem do panorama atual. PMI Journal – PMI-RS 3, pp. 7-16.
- Wideman R. M. (2002). Comparing PRINCE2® with PMBoK®. Disponível em: <<http://www.pmforum.org/library/papers/Prince2vsGuide3easrd1.htm>>. Acessado em: 01 de abr. 2004.

Capítulo 12

Comment [T280]: Está fora do padrão do livro. Verifica o template

Gestão de Riscos Em em Projetos de Software

Comment [T281]: É para ser tamanho 20

Comment [T282]: É para ser tamanho 14

Neste capítulo buscamos identificar os riscos explícitos no desenvolvimento de um projeto de software estabelecendo mecanismos aos gerentes de projetos adiantar-se aos fatores negativos que podem comprometer o andamento de suas atividades do projeto e apoiar-se os fatores positivos, diminuindo as possibilidades dos riscos. Esta unidade abrange a introdução aos riscos, os principais processos na gestão de riscos, além de sugestões de leituras, tópicos de pesquisas e exercícios. Este capítulo direciona-se as questões relevantes aos riscos, demonstrando a importância da gestão de riscos para um projeto de software, quais os principais fatores que podem contribuir para o fracasso do projeto e os desafios enfrentados pela gestão de riscos. Cada processo é minuciosamente detalhado e estudado nas seções seguintes.

Comment [T283]: Você deve ser impessoal no Capítulo, conforme sugerido pelos professores. Deve ser usado a 3ª pessoa do singular.**Comment [T284]:** Nesse capítulo já identifica os riscos no desenvolvimento ou dá meios para o leitor identificar?**Comment [T285]:** Ficaria melhor: Para que os**Comment [T286]:** Ficaria melhor: possam adiantar-se**Comment [T287]:** nos**Comment [T288]:** Não está legal chamar o capítulo de : Esta unidade. Ou você junta o que tem no próximo período com esse, ou escreve de outra forma.

12.1. Gestão de Riscos

Comment [T289]: Fonte 14 Negrito

“Atualmente diversas empresas estão se dando conta da importância de gerenciar os seus projetos de software dentro de técnicas comprovadamente eficientes e de metodologias sistêmicas. Como consequência consequência, os resultados obtidos são expressivos quando os gestores (gerentes de projetos) atentam para a gestão dos riscos envolvidos, minimizando seu impacto e exposição a estes ou, ainda que ocorram, mas de forma controlada, ou seja, aceitos somente quando potenciais benefícios e probabilidades de sucesso preponderarem sobre os custos de fracasso e/ou perda da satisfação. Dentro deste enfoque a mitigação tem se mostrado uma estratégia eficaz de resposta aos riscos nos projetos de software” [SILVA 2007].

Comment [T290]: Não tem esse espaço antes do primeiro parágrafo.

A gestão de riscos em projetos de software consome grande parte das atribuições do gestor, que pretende que os riscos não ocorram ou ainda seja tornar mínimo ao máximo sem trazer grandes prejuízos ao cronograma, estourar os custos do projeto ou o prazo para entrega, etc. Vieira [1] afirma com propriedade que "o gerenciamento dos riscos é o trabalho principal de uma gestão de projetos" e Verzuh [2] mais enfaticamente discorre: "toda gestão de projeto é um gerenciamento dos riscos".

Comment [T291]: Não entendi esse trecho aqui**Comment [T292]:** Referência fora do padrão**Comment [T293]:** Referência fora do padrão

Estas afirmações são verdadeiras no dia-a-dia de qualquer gestor ao passo que este tem como premissas fundamentais esta identificando os possíveis riscos antes que estes ocorram, provendo mecanismos que permitam minimizar ou eliminar a probabilidade e impactos nos seus projetos [SILVA

Comment [T294]: ????

2007]. Importante salientar que todo risco mesmo sendo baixa baixo existe uma possibilidade mínima de ocorrência, logo sua ação nem sempre é certa; mais dependendo do seu grau de incidência X ocorrência pode afetar negativamente o projeto ou pode abrir uma grande oportunidade de negócio [WIDERMAN 2003].

Comment [T295]: ????

Muitos estudos voltados para o tratamento de risco tem têm sido publicado desde 1989 (Pfleeger [4], Boehm[5], Kontio e Basili[6], Keil[7]), o que mostra a tese de que o assunto gestão de riscos não se trata de uma novidade do mercado de software. Umas pesquisas realizadas por órgãos de competência internacional comprovam que a maioria das falhas em projetos de software nas organizações está associada à falta de habilidade de identificar e gerenciar riscos.

Comment [T296]: Referência fora do padrão;

Comment [T297]: Esse artigo não está legal aqui.

Por muitos anos, desenvolvedores, e engenheiros de software, se preocupavam em somente entender o problema do cliente e logo em seguida programar, sem se preocupar com nenhum tipo de incerteza que poderia ser encontrada no desenvolvimento que poderia causar transtornos ao seu projeto.

Comment [T298]: Repetido, coloca: causaria, fica melhor.

Comment [T299]: Ou você tira essa palavra COM, ou você muda possibilitou para foi possibilitada.

Com a nova visão que surgiu em torno da tecnologia da informação, possibilitou uma modalidade diferenciada de profissionais (gerentes de negocio negócio, ou de projeto de TI). esses profissionais se preocupam com todos os processos do projeto, análise análise, planejamento de todos os processos para o cumprimento das metas do projeto, inclusive os produtos intermediários a cada momento susceptíveis suscetíveis a riscos [SILVA 2007].

Comment [T300]: Aqui seria ponto final.

Comment [T301]: E aqui seria letra maiúscula.

“Recente pesquisa publicada pela Chaos[8] (1999 e 2001) apresentou, após análise análise de 30.000 projetos de aplicações em tecnologia da informação, uma crescente evolução no gerenciamento de projetos através da conseqüente consequente gestão de riscos” [SILVA 2007]

Comment [T302]: Referência fora do padrão.

Comment [T303]: Não acho que isso seja recente mais...

O PMBOK dividi divide a administração dos riscos do projeto em (1 – planejamento do gerenciamento dos riscos, 2 - identificação dos riscos, 3 – análise qualitativa dos riscos, 4 – análise quantitativa dos riscos, 5 – planejamento de respostas a riscos e 6 – monitoramento e controle de riscos) brilantemente explicado em Oda[9]; todavia focando especificamente no tratamento destes riscos, observamos mediante experiências vividas em projetos nesta área que a mitigação tem se mostrado uma das melhores práticas na resposta aos riscos que surgem.

Comment [T304]: Essa palavra Brilantemente está esquisita aqui. E a referência está fora do padrão.

Comment [T305]: Deve ser impessoal.

Apresentaremos a seguir os principais processos utilizados pela gestão de riscos em projetos de software baseado no PMBOK.

Comment [T306]: Deve ser impessoal. Nada de 1ª pessoa do plural.



Comment [T307]: Figura não pode ser colorida. Deve ser em tons de cinza. Deve estar centralizada.

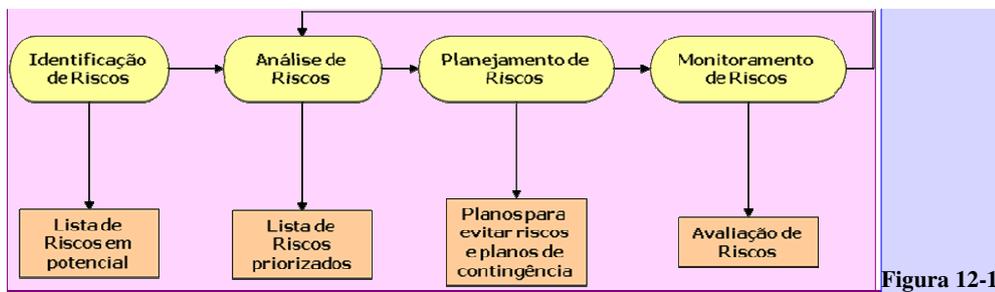


Figura 12-1.

Processos do Gerenciamento de riscos

Fonte: Gestão de Riscos

Comment [ajhol308]: A figura não está coerente com o processo de gerenciamento de riscos mais difundido (vide figura abaixo).

Comment [T309]: Deve ser citada a referência da fonte

Comment [T310]: Não está de acordo com o padrão. Verificar no template. A numeração e a legenda estão fora do padrão.

12.2. Análise Análise e Reuniões de Planejamento de Riscos

As equipes dos projetos organizam reuniões para o desenvolvimento do planejamento de riscos. Nessas reuniões participam: Gerentes de projeto, membros da equipe do projeto e pessoas interessadas ao projeto, outras pessoas da organização com funções de gerenciamento das atividades e execução e planejamento de riscos, e outras pessoas quando forem necessários.

O plano básico para executar as atividades de gerenciamento de riscos é definido nessas reuniões. Serão desenvolvidos os elementos de custo de riscos e as atividades do cronograma de riscos para serem incluídos no orçamento e cronograma do projeto, respectivamente. Serão designadas as responsabilidades de riscos. Modelos organizacionais gerais para categorias de risco e definições de termos como níveis de risco, probabilidade por tipo de risco, impacto por tipo de objetivos, além da matriz de probabilidade e

impacto, serão adaptados para o projeto específico. As saídas dessas atividades serão definidas no planejamento de gerenciamento de riscos.

12.3. Planejamento do Gerenciamento de Riscos

No processo de planejamento do gerenciamento de riscos são adotadas estratégias de abordagem, planejamento e execução das atividades do gerenciamento dos riscos em um projeto de software.

Essa atividade é de suma importância para o seu projeto, pois ela assegura que o nível, o tipo e a visibilidade do gerenciamento de riscos se enquadre com os riscos e a importância do projeto em relação a empresa, dessa forma teremos mecanismos para fornecer tempo e recurso suficientes para permitir a execução das atividades de gerenciamento de riscos e subsídios para fornecer alternativas para o controle dos riscos.

Descreveremos logo abaixo alguns requisitos fundamentais que devem ser abordados pelo planejamento do gerenciamento de riscos.

- **Metodologia** – define a abordagem, as ferramentas e as fontes de dados utilizadas durante a fase de gerenciamento de riscos.
- **Funções e Responsabilidades** - Definem liderança, suporte e participação da equipe de gerenciamento de riscos em cada tipo de atividades do plano de gerenciamento de riscos. Distribuem funções aos integrantes do time, tiram as dúvidas quanto à responsabilidade de cada membro envolvido.
- **Orçamentação** – Distribue recursos para o planejamento do gerenciamento de riscos e estima-se os custos necessário para o gerenciamento, sendo que esses custos devem ser incluído nos custos do projeto.
- **Tempos** – avalia quando e com que frequência o gerenciamento de riscos será executado durante todo o projeto além disso define quais as atividades serão incluídas no cronograma do projeto.
- **Categorias dos Riscos** - Fornece uma estrutura que garante um processo abrangente para identificar sistematicamente os riscos até um nível consistente de detalhes e contribui para a eficácia e qualidade da identificação de riscos. Uma organização pode usar se uma estrutura similar a essa para categorizar seus riscos, e novas categorias dos riscos podem ser incluídos incluídos no decorrer do projeto, visto que esse processo é contínuo. Uma boa dica seria revisar as categorias dos riscos antes do processo de planejamento, ou seja,

Formatted: Indent: First line: 0"

Comment [T311]: Deve ser impessoal.

Formatted: Font: Not Bold

antes de usá-la no processo de identificação dos riscos, logo abaixo podemos observar melhor através da figura [Figura 12-2](#) a seguir.

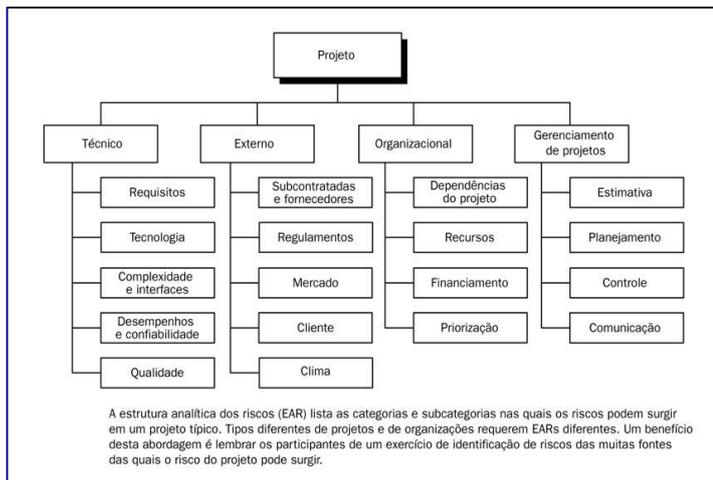


Figura 12-2. Exemplo de uma estrutura analítica dos riscos (EAR)

Fonte: Estrutura Analítica dos Riscos

- **Definições de Probabilidade e Impacto dos Riscos** – A qualidade e credibilidade do Processo de Análise [Análise Qualitativa](#) de riscos exigem a definição de níveis diferentes de probabilidade e impactos de riscos. As métricas de riscos e a proporção do impacto são adequados ao projeto individual durante o processo de planejamento do gerenciamento de riscos.

Você poderia ainda tomar por base uma escala, essa escala poderia variar dos riscos que raramente poderia acontecer caracterizando eles como “muito improvável” até aqueles riscos com alta chance dele ocorrer como “altamente provável”, feito isso voce definiria uma escala numérica para avaliar a probabilidade desses riscos acontecerem durante o seu projeto por exemplo uma escala (0,1; 0,3; 0,5; 0,7; 0,9), onde cada valor representa uma escala do risco ocorrer.

A escala de impacto é muito importante para o projeto, pois avalia o grau de importância do impacto, como sendo negativa para as ameaças ao projeto, ou seja, os riscos que podem afetar o desenvolvimento do seu projeto e positiva para as oportunidades que possam surgir em cada fase do projeto se o risco ocorrer. As escalas de impacto são específicas do objetivo potencialmente afetado, do tipo e do tamanho do projeto, da situação financeira, das estratégias da organização e da sensibilidade da

Comment [T312]: O que está escrito na figura está ruim de ser lido. Ela deve ser centralizada. Verificar padrões SBC.

organização a impactos específicos. “As escalas relativas de impacto são descritores classificados de forma simples, como “muito baixo”, “baixo”, “moderado”, “alto” e “muito alto”, refletindo impactos cada vez maiores conforme definidos pela organização” [FERRARI 2006]. Os valores das escalas refletem-se nos valores do impacto no projeto. Essas escalas podem receber valores lineares (0,1, 0,3, 0,5, 0,7, 0,9) ou não-lineares (0,05, 0,1, 0,2, 0,4, 0,8). As escalas não-lineares podem representar o desejo da organização de evitar ameaças de alto impacto ou de explorar oportunidades de alto impacto, mesmo se elas tiverem uma probabilidade relativamente baixa. No uso de escalas não-lineares é importante entender o significado dos números e como se relacionam entre si, como são derivados e o efeito que podem ter sobre os diversos objetivos do projeto.

A Figura 12-3 é um exemplo de impactos negativos de definições que poderiam ser usadas na avaliação dos impactos de riscos relacionados a quatro objetivos do projeto. Essa figura demonstra a abordagem relativa e numérica em uma única figura, deixando claro que não estamos comparando as duas abordagens, mais sim apresentando.

Condições definidas para escalas de impacto de um risco em objetivos importantes do projeto (os exemplos são mostrados somente para impactos negativos)					
Objetivo do projeto	São mostradas escalas relativas ou numéricas				
	Muito baixo / 0,05	Baixo / 0,10	Moderado / 0,20	Alto / 0,40	Muito alto / 0,80
Custo	Aumento de custo não significativo	Aumento de custo < 10%	Aumento de custo de 10% a 20%	Aumento de custo de 20% a 40%	Aumento de custo > 40%
Tempo	Aumento de tempo não significativo	Aumento de tempo < 5%	Aumento de tempo de 5% a 10%	Aumento de tempo de 10% a 20%	Aumento de tempo > 20%
Escopo	Diminuição do escopo quase imperceptível	Áreas menos importantes do escopo afetadas	Áreas importantes do escopo afetadas	Redução do escopo inaceitável para o patrocinador	Item final do projeto sem nenhuma utilidade
Qualidade	Degradação da qualidade quase imperceptível	Somente as aplicações mais críticas são afetadas	Redução da qualidade exige a aprovação do patrocinador	Redução da qualidade inaceitável para o patrocinador	Item final do projeto sem nenhuma utilidade

Esta tabela apresenta exemplos de definições de impactos de riscos para quatro objetivos diferentes do projeto. Elas devem ser adequadas no processo Planejamento do gerenciamento de riscos ao projeto individual e aos limites de risco da organização. As definições de impactos podem ser desenvolvidas de forma semelhante para as oportunidades.

Comment [T313]: Leitura ruim da escrita da figura. Deve estar centralizada. Não está padronizada na legenda.

Figura 12-3. Definição de escalas de impacto para quatro objetivos do projeto

Fonte: Impacto dos riscos

12.4. Identificação dos Riscos

Peter Drucker [DRU75 2004] disse certa vez: “Embora seja fútil tentar eliminar o risco e questionável tentar minimizá-lo, é fundamental que os riscos assumidos sejam os riscos certos.” O mais importante dentro de um projeto de software antes de tudo é a identificação de todos os riscos que possam

comprometer seu produto final, para depois avaliar quais seriam os riscos que podemos considerá-los “certos”.

A identificação de riscos determina os riscos que podem afetar o andamento do projeto e relaciona suas características. Os membros da equipe do projeto podem participar dessa atividade quando for solicitada a presença dos envolvidos, desde o gerente de projeto, os membros da equipe do projeto, pessoas envolvidas no gerenciamento de riscos, especialistas, clientes, usuários finais e pessoas interessadas diretamente na execução do projeto. Todas as pessoas envolvidas no projeto devem ser impulsionadas a participarem da atividade de identificação de riscos, quanto maior o comprometimento de todos, maior o sucesso de seu projeto.

A identificação dos riscos é um processo interativo, pois podem ser encontrados novos riscos durante o andamento do projeto. As interações e os participantes podem variar de caso a caso. Os membros do projeto devem ser submetidos a todas as interações para manter um grau de comprometimento e responsabilidade relacionados aos riscos e ações de respostas a riscos. Outras pessoas que não fazem parte do projeto também podem ser fornecer dados que podem auxiliar na gestão de riscos no processo de identificá-los.

12.4.1. Classificação dos Riscos

Os riscos podem ser classificados de acordo a sua natureza em: **riscos técnicos, riscos de projeto, riscos de processos e riscos de negócio.**

- **Riscos Técnicos** – os riscos técnicos afetam diretamente a qualidade do seu produto final, prejudicando a conclusão do projeto.. Os riscos técnicos se relacionam com a tecnologia na qual esta está sendo implementado o seu software. Podemos fazer uma avaliação dos riscos técnicos antes de começar com alguns questionamentos como: a tecnologia na qual vai ser implementado é de conhecimento de toda a equipe de desenvolvimento? Não há necessidade de capacitações para o desenvolvimento nessa tecnologia? A tecnologia que vamos desenvolver já foi usada em outros projetos de sucesso? Alguns questionamentos como esses podem evitar o surgimento de riscos técnicos. técnicos.

- **Riscos de Projeto** - Identifica os riscos ligados aos aspectos organizacionais, operacionais e contratuais ameaçando o plano do projeto, causando prejuízo ao cronograma e custos do projeto. Os riscos de projeto estão relacionados ao uso de recursos quanto a isso podem ser subdivididos Organizacionais, humanos e tempo.

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Formatted: Justified, Indent: Left: 0,43", Hanging: 0,01", Space After: 0 pt, Bulleted + Level: 1 + Aligned at: 0,27" + Indent at: 0,52"

Comment [T314]: Padronizar tamanho do travessão.

Formatted: Font: Not Italic

- **Riscos de Processos** – São localizados no planejamento do projeto, na aquisição de recursos humanos, ao longo do projeto no controle e acompanhamento e na segurança da qualidade.

Formatted: Font: Not Italic

- **Riscos de Negócio** – O risco de negócio é considerado o mais crítico dos riscos, pois ele pode destruir todo o seu planejamento e principalmente pode abortar o seu projeto. Ele ameaça a distribuição ou a implantação do produto do seu projeto, prejudicando o retorno do investimento feito. São vários os riscos de negócio que podem surgir podemos destacar alguns: o que aconteceria se a concorrência lançasse um produto similar o ao seu? O valor do produto é maior que o custo do projeto? Você realizou uma pesquisa de mercado para verificar se o projeto é viável? Os vendedores conhecem o produto, sabem como vendê-los? São algumas dessas situações que podem ocasionar um risco de negócio.[negócio.](#)

Formatted: Font: Not Italic

Formatted: Font: Bold, Italic

-
-

Formatted: Font: Not Bold

Formatted: Font: (Default) Times, 12 pt

Formatted: Heading 3 Char;PSC_Titulo_3 Char1, Justified, Indent: Left: 0,43", Hanging: 0,01", Don't add space between paragraphs of the same style, Bulleted + Level: 1 + Aligned at: 0,27" + Indent at: 0,52"

12.4.2. Métodos para Coleta de Dados para Identificação dos Riscos

Formatted: Font: Italic

Quando desejamos agrupar dados para identificar possíveis riscos referentes ao projeto utilizam-se algumas técnicas, logo a baixo descreveremos algumas delas:[delas:](#)

Comment [T315]: Deve ser impessoal.

Formatted: Indent: First line: 0"

12.4.2.1. Revisões da documentação

Pode ser feita uma revisão organizada, estruturada de toda a documentação do projeto, abrangendo planos, premissas, arquivos de projetos anteriores além de outras fontes de informações. A consistência entre os planos e com as premissas e os requisitos do projeto podem indicar futuros riscos.[riscos.](#)

Formatted: Indent: First line: 0"

12.4.2.2. Técnicas de Coleta de Informações

Algumas técnicas podem ser utilizadas na coleta de informações para serem utilizadas na identificação de riscos, essas podem incluir:

- **Brainstorming** – O objetivo dessa técnica é a aquisição de uma lista de riscos do projeto. Normalmente ela é empregada por um conjunto de especialistas que não estão ligados ao projeto. Porém, utiliza-se um mediador para demonstrar ideias sobre os riscos do projeto. Pode utilizar a classificação dos riscos como referência.

- **Técnicas Delphi** – A meta dessa abordagem é encontrar o ponto comum entre os especialistas. Nessa técnica um facilitador distribui um questionário entre os especialistas solicitando idéias sobre os riscos importantes ao projeto. As opiniões são resumidas e depois discutidas a fim de realizar comentários extras, pontos relevantes. Essa técnica é interessante, pois diminui as chances de alguma pessoa possa indevidamente influenciar no resultado final.
- **Entrevistas** – As entrevistas entre especialistas, pessoas ligadas diretamente ao projeto, como a gerência, o usuário, as pessoas experientes, as pessoas interessadas no produto final, podem também de forma segura identificar possíveis riscos do seu projeto.
- **Análise dos Pontos Fortes e Fracos, Oportunidades e Ameaças (SWOT)** – “Através dessa técnica certifica-se que o a análise realizada no projeto da cada uma das expectativas da análise SWOT, isso faz com que aumente amplitude dos riscos”[FERRARI 2004].

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Comment [T316]: Padronizar com o tamanho igual aos anteriores.

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Comment [T317]: E o que é análise SWOT? Cabe aqui uma explicação sobre ela, descrevendo-a.

12.4.2.3. Análise da Lista de Verificação

A lista de verificação de identificação de riscos pode ser elaborada a partir das informações de informações históricas e no conhecimento adquirido em outros projetos similares a esse e de outras fontes que de informações.

Formatted: Indent: First line: 0"

O nível inferior da **Estrutura Analítica dos Riscos (EAR)** pode ser utilizado como uma lista de verificação de riscos. Geralmente essa lista de verificação de riscos é simples e rápida, portanto se torna inviável construir uma lista onde aborda todos os riscos, por isso a necessidade também de explorar outros riscos que não aparecem na lista de verificação.

12.4.2.4. Análise das Premissas

“Qualquer projeto de software é desenvolvido a partir de um conjunto de: hipóteses, idéias, cenários ou premissas” [FERRARI 2004]. Esta etapa tem por finalidade validar as premissas de acordo com sua aplicação no projeto. Ela identifica os riscos do projeto causados por efeitos impróprios, inconsistentes ou incompletos das premissas.

Formatted: Indent: First line: 0"

12.4.2.5. Técnicas com Diagramas

As técnicas com diagramas podem ser utilizadas no gerenciamento de riscos da seguinte forma:

- **Diagrama de Causa Efeito** – Realizado para controle de qualidade, esse diagrama. Também chamado de diagramas de *Ishikawa* ou diagramas espinha de peixe, demonstra como os diversos fatores podem estar ligados a possíveis problemas, vejamos a baixo na figura [Figura 12-4](#) um exemplo de um diagrama de causa efeito.

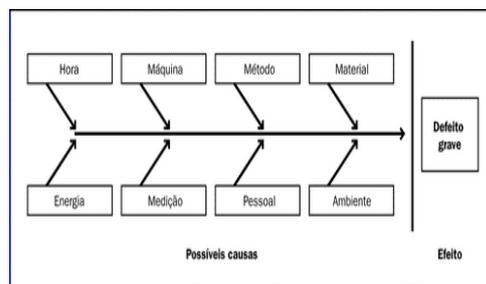


Figura 12-4- Diagrama de causas efeito

Fonte: Identificação dos Riscos

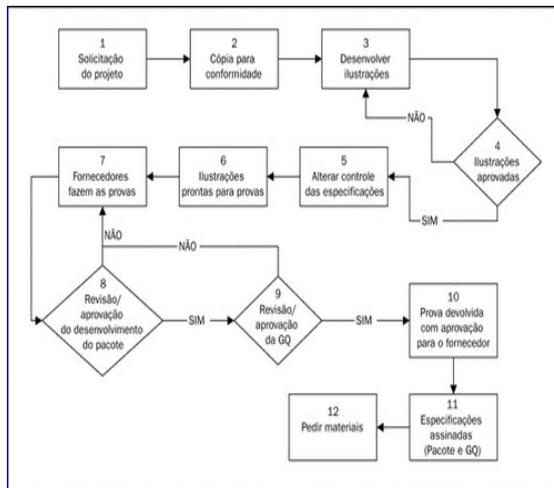
- **Diagrama do Sistema ou Fluxograma** – auxilia no processo de detalhamento do problema, mostrando como eles ocorrem. Existem diversos fluxogramas, mais em sua maioria demonstra as atividades, os pontos de decisão e a ordem de processamento. Além disso, eles detalham o inter-relacionamento com todos os processos envolvidos no projeto. Na [figura 12-5](#), mostra um fluxograma utilizado na revisão de projetos. Podem ser notados nos fluxogramas alguns problemas de qualidade que podem ocorrer, e aonde eles podem surgir.

Formatted: Indent: First line: 0"

Formatted: Font: Italic

Comment [T318]: Deve ser impessoal.

Comment [T319]: Melhorar a escrita da figura ou aumentar a figura.



Comment [T320]: Letras da figura ilegíveis. Figura deve ser centralizada.

Figura 12-5 – Diagrama de Fluxograma

Fonte: Identificação dos Riscos

12.5. Análise Qualitativa de Riscos

Comment [T321]: Tem alguma referência?

Análises qualitativas dos riscos incluem mecanismos que permitem priorizar riscos identificados para ações futuras ligadas aos riscos como análise quantitativa dos riscos e planejamento as respostas dos riscos. As empresas tendem a alcançar excelentes resultados em seus projetos quando se concentram nos riscos do projeto de alta prioridade. Este processo da gestão de riscos realiza avaliação dos riscos identificados de acordo com a probabilidade dos riscos ocorrerem e com o impacto causado no escopo do projeto.

Formatted: Indent: First line: 0"

Análise qualitativa dos riscos é uma forma objetiva, simples, rápida e de pequeno custo para estabelecimento das prioridades para o planejamento das respostas aos riscos, além de instituir métricas para análise quantitativa dos riscos, esse processo só é utilizado quando solicitada pela equipe de gerenciamento de riscos.

12.5.1. Avaliação de Probabilidade e Impacto dos Riscos

Nesta fase da análise qualitativa dos riscos, empregam-se métodos para avaliação de probabilidade caso os riscos venham a ocorrer. Avaliação do impacto do risco ocasionado elabora os efeitos sentidos no escopo do projeto, no objetivo: Aspectos financeiros relacionado aos custos, qualidades, além das ameaças os transtornos causados pelos riscos e também os riscos que possam trazer oportunidades abrir espaço para novas idéias.

Formatted: Indent: First line: 0"

A probabilidade e a força dos riscos são avaliados individualmente para cada risco identificado. Eles podem ser avaliados de diversas formas desde simples reuniões com sua equipe, com entrevistas com membros do projeto, até as sugestões de especialistas não ligados ao projeto. A avaliação de terceiros, ou seja, de especialistas não desligados do projeto se faz necessário, pois, em muitos casos há poucos dados sobre riscos na sua base de dados. Sendo assim a opinião de especialistas ajudam nesse processo, visto que muitos participantes não possuem experiência alguma com riscos, então especialistas podem conduzir a equipe para avaliação dos riscos.

“A probabilidade de cada risco e seu impacto em cada objetivo são avaliados durante a entrevista ou reunião. Os detalhes da explanação, inclusive as premissas que justificam os níveis atribuídos, também são registrados” [FERRARI 2004]. As probabilidades e os impactos podem ser considerados de acordo com os definidos na fase de planejamento do gerenciamento de projetos. Em muitas situações, os riscos, com pouca probabilidade de ocorrer e com impacto baixo ao projeto, não são levados em consideração, mas mesmo nesses casos os riscos devem ser monitorados e controlados.

12.5.2. Matriz de Probabilidade e Impacto

Formatted: Indent: Left: 0"

“Os riscos podem ser priorizados para análise quantitativa e respostas adicionais, com base na sua classificação. As classificações são atribuídas aos riscos com base em sua probabilidade e impacto avaliados. A avaliação da importância de cada risco e, portanto, a prioridade da atenção é normalmente realizada usando uma tabela de pesquisa ou uma matriz de probabilidade e impacto” [FERRARI 2004]. Vejamos a seguir um exemplo usando uma matriz de probabilidade e impacto dos riscos identificado em um projeto, onde temos uma tabela para medir a probabilidade de ocorrência a partir de valores e suas chances de acontecerem e uma tabela onde podemos avaliar o impacto causado ao projeto com esse risco identificado. Dessa forma construiremos a matriz relacionando a probabilidade do risco e o impacto causado no projeto, onde teremos uma pontuação final para cada risco.

Comment [T322]: Deve ser impessoal.

Comment [T323]: Deve ser Impessoal, não pode ser assim.

Comment [T324]: Deve ser impessoal.

Comment [T325]: Não pode primeira pessoa

Comment [T326]: Mesma coisa

1 - Analise Qualitativa das Probabilidades	
Referencial	Probabilidade de Ocorrência
Grande chance ocorrer	0,95
Possivelmente ocorrerá	0,75
Mesma chance de ocorrer ou não	0,5
Baixa chance de ocorrer	0,25
Mínima chance de ocorrer	0,10

Comment [T327]: Cadê a legenda da tabela?

2 - Avaliações do Impacto	
Grau do Impacto	Peso
Extremamente grande	5,0
Grande	4,0
Mediana	3,0
Pequeno	2,0
Bem pequeno	1,0

Comment [T328]: Cadê a legenda da tabela

Matriz de Probabilidade e Impacto

	Pontuação Para Cada Risco Identificado					
Probabilidade						

Comment [T329]: Tabela descrita fora do padrão. Deve ter numeração da tabela. Fonte, Legenda sem ser em negrito. Tamanho 12.

0,95	0,95	1,90	2,85	3,80	4,75	
0,75	0,75	1,50	2,25	3,0	3,75	
0,5	0,50	1,0	1,50	2,0	2,5	
0,25	0,25	0,50	0,75	1,0	1,25	
0,1	0,10	0,20	0,30	0,40	0,50	
	1,0	2,0	3,0	4,0	5,0	
	Impacto					

Comment [T330]: Melhorar essa tabela. Está mal feita. Com tantas células vazias.

Com base na matriz de probabilidade e impacto, podemos priorizar os Riscos do Projeto como:

- **Baixo Risco: 0,10 a 0,75**
- **Médio Risco: 0,95 a 1,90**
- **Alto Risco: 2,0 a 4,75**

12.5.3. Avaliação da Qualidade dos Dados Sobre os Riscos

Uma análise realizada de forma qualitativa sobre riscos requer informações consistentes, corretas, exatas e bastante confiáveis. Esta avaliação é uma técnica utilizada para avaliar a natureza da utilidade dos riscos para o gerenciamento dos riscos. Ela envolve métodos para analisar até que ponto a ameaça é percebida e o grau de qualidade, de confiabilidade, de precisão e de integridade desses eventos. Informação com pouca qualidade implica em uma análise qualitativa de riscos de pouca utilidade ao projeto. Por isso foi adotado numa etapa anterior algumas técnicas de coleta de informações. Essa atividade requer tempo e muita paciência dos envolvidos, pois os dados podem afetar diretamente ao seu projeto final.

Formatted: Indent: First line: 0"

12.5.4. Categorização dos riscos

“Os riscos do projeto podem ser categorizados por fontes de risco (por exemplo, usando a EAR), pela área do projeto afetada (por exemplo, usando a EAP) ou por outra categoria útil (por exemplo, fase do projeto) para determinar as áreas do projeto mais expostas aos efeitos da incerteza. O agrupamento dos riscos por causas-raiz comuns pode possibilitar o desenvolvimento de respostas a riscos eficazes” [FERRARI 2004].

Comment [T331]: Acho que caberia uma definição para cada categorização, aqui.

Formatted: Indent: First line: 0"

12.5.5. Avaliação da Urgência do Risco

Dentre os riscos identificados sempre haverá alguns tratados com maior prioridade do que outros, e Esses riscos exigem resposta em curto exigindo urgência na sua solução para que não venha afetar o projeto.

Formatted: Indent: First line: 0"

Comment [T332]: ????

Alguns fatores podem indicar prioridade dos riscos na avaliação como podemos citar: o tempo necessário para apresentar uma solução aos riscos, os sintomas apresentado pelos riscos, a emissão de sinais de alertas e principalmente a classificação dos riscos.

12.6. Análise Quantitativa de Riscos

Este processo tem por finalidade avaliar numericamente a possibilidade de cada ameaça ocorrer e suas possíveis consequências junto ao objetivo do projeto. Este processo usa-se de algumas técnicas como:

Formatted: Indent: First line: 0"

simulação de Monte Carlo e análise de decisão para:

Comment [T333]: É bom ter uma definição para cada uma delas.

- Assimilar custos, orçamentos, cronogramas ou objetivos reais e alcançáveis.
- Identificar as ameaças potenciais, expondo numericamente sua contribuição relativa aos riscos do projeto.
- Classificar o grau de probabilidade em se atingir um objetivo relativo ao projeto.
- Quantificar o grau de exposição de um risco para o projeto.
- Relacionar o tamanho da reserva do custo e cronograma que pode ser necessária.

Assim com na análise qualitativa dos riscos a abordagem de análise quantitativa dos riscos requer antes de tudo a identificação dos riscos. As duas análises podem ser realizadas paralelamente ou individualmente. “Considerações com relação à disponibilidade de tempo e orçamento e a necessidade para declarações qualitativas ou quantitativas sobre risco e impactos determinarão que método(s) usar. Tendências nos resultados quando a análise quantitativa é repetida pode indicar a necessidade de mais ou menos ação de gerenciamento de risco.” [PMBOK, 2004].

12.6.1. Técnicas de Representação e Coletas de Dados

- **Entrevistas** – “As técnicas de entrevistas são usadas para quantificar a probabilidade e o impacto dos riscos nos objetivos do projeto. As informações necessárias dependem do tipo de distribuições

de probabilidades que será usado.” [FERRARI, 2004]. Podemos observar o seguinte: Quando desejamos reunir dados do projeto, vamos considerar as informações de uma situação mais otimista, ou seja, os riscos cujos níveis de evidência são baixos, ou numa situação pessimista, ou seja, aqueles riscos são fortes candidatos a ocorrer e prejudicar seu projeto e mais provável para algumas distribuições comumente usadas, e a média e o desvio padrão para outras. Exemplos de estimativas de três pontos para uma estimativa de custos são mostrados na Figura 12-7. Todas as informações do projeto, como a documentação da análise lógica da fase de riscos, e informações adicionais sobre eventos adversos, são subsídios de extrema relevância para as entrevistas sobre gerenciamento de riscos, nas documentações pode haver dados confiáveis e de credibilidade para análise.

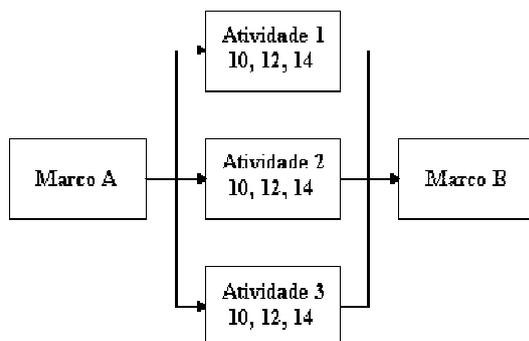


Figura: 12-7 - Estimativa em 3 pontos para estimativa de custos.

Fonte: PMBOK

- **Distribuições de Probabilidade** – esta etapa do processo de análise quantitativa dos riscos representa a insegurança das informações, a confiabilidade dos dados com o tempo previsto no cronograma e os custos nos artefatos do projeto. As distribuições importam ao mesmo tempo à perspectiva e às decorrências dos itens do projeto. Alguns tipos comuns de distribuições são: **uniforme, normal, triangular, beta e log normal.**

A Figura 12-8 mostra duas das distribuições, onde o eixo vertical representa a expectativa e o eixo horizontal o impacto.

- Valor Monetário Esperado (EMV em inglês) do resultado = Resultado x Probabilidade daquele resultado.

Comment [T334]: Não pode ser usada a primeira pessoa

Comment [T335]: Primeira pessoa

Comment [T336]: Primeira pessoa

Comment [T337]: No plural mesmo?

Comment [T338]: Fora do padrão. Deve ser 12.7

Comment [T339]: Fora do padrão. Não deve estar em negrito. Deve ser fonte 12. Deve ter referência na fonte.

Formatted: Indent: First line: 0,5"

- Valor Monetário Esperado de uma decisão = somas dos EMVs de todos os resultados derivados daquela decisão.
- O cronograma agressivo tem um valor monetário esperado de \$4.000 e é "preferido" em relação ao Cronograma conservador que tem um EMV de \$1.000.

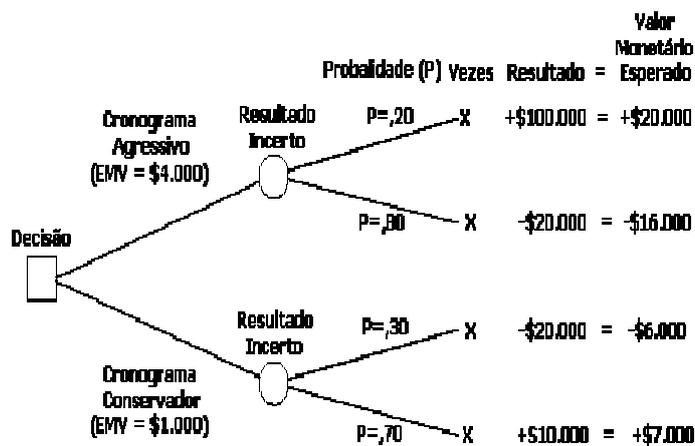


Figura: 12-8, Árvore de decisão

Fonte: PMI

12.6.2. Análise quantitativa de riscos e técnicas de modelagem

Os métodos utilizados pela análise quantitativa dos riscos abrangem:

- **Análise Análise Sensitiva** – essa etapa auxilia o gerenciamento de riscos na identificação dos riscos com maior impacto no objetivo do projeto. “Ela examina a extensão com que a incerteza de cada elemento do projeto afeta o objetivo que está sendo examinado quando todos os outros elementos incertos são mantidos em seus valores de linha de base” [FERRARI 2004].
- **Análise da Árvore de Decisão** – “Uma análise de decisão é normalmente estruturada como uma árvore de decisão. A árvore de decisão é um diagrama que descreve uma decisão sob consideração e as implicações de escolher uma ou outra das alternativas disponíveis” [PMI, 2004]. Ela anexa às [as](#) perspectivas dos riscos e os custos de recompensas de cada abertura lógica dos eventos e de decisões futuras. [Solucionando a árvores de decisão](#) indicará quais decisões geram as estimativas

Comment [T340]: Fica melhor: Ao solucionar ... serão indicadas quais decisões...

maiores para o tomador de decisão quando todas as consequências incertas, os custos, as recompensas e as decisões seguintes são **forem** quantificadas.

- **Simulação** – “Uma simulação do projeto usa um modelo que traduz as incertezas especificadas em um nível detalhado para o impacto potencial delas nos objetivos que são expressos no nível do projeto total. Simulações do projeto são tipicamente executadas usando a técnica do Monte Carlo” [FERRARI]. Na simulação o escopo do projeto é avaliado diversas vezes, cujos valores iniciais são calculados aleatoriamente a partir de uma função de probabilidade selecionada durante cada interação.

12.7. Planejamento de Respostas a Riscos

São adotadas várias estratégias utilizadas para resposta a riscos. Para cada risco identificado deve ser selecionada uma estratégia ou associação de estratégias com maior probabilidade na sua eficácia [PMI 2004]. Algumas ferramentas estudadas acima como análise da árvore de decisão pode ser uma ótima ideia para escolher a resposta mais adequada aos riscos. Em seguida, são desenvolvidas ações específicas para implementar essa estratégia. Podem ser utilizadas estratégias principais e de reservas. É possível desenvolver um plano alternativo para ser implementado se a estratégia selecionada não for totalmente eficaz ou se um risco aceito ocorrer [FERRARI 2004]. Por fim, os planos de contingências podem ser desenvolvidos juntamente com a identificação das condições que provocaram a sua execução. Mostraremos abaixo as principais estratégias adotadas no Planejamento de Respostas a Riscos.

12.7.1. Estratégia para Riscos Negativos ou Ameaças

Três estratégias são adotadas para riscos negativos ou para ameaças que podem comprometer os objetivos do seu projeto: Prevenir, Transferir ou Mitigar. Cada uma dessas estratégias **seria** abordada logo em seguida.

- **Evitar** - Evitar o risco é mudar o plano de projeto para eliminar o risco ou a condição ou para proteger os objetivos do projeto destes impactos. Embora a equipe não possa eliminar todos os eventos de risco, alguns riscos específicos podem ser evitados [PMBOK 2004]. **Alguns riscos** quando identificados logo cedo podem ser impedidos de **agirem com esclarecimentos**, obtendo mais informações a respeito da ação do risco ou até mesmo

Formatted: Indent: First line: 0"

Comment [T341]: Troca o seria por é

Comment [T342]: Evitar ou Prevenir. Procura ser coerente com o que você citou acima.

Comment [T343]: Escreve com outras palavras, porque já tem nessa mesma linha e final da anterior as palavras alguns riscos.

Comment [T344]: Como assim??

consultando um especialista, tudo isso para reduzir sua ação, a fim de isolar seus objetivos do projeto do impacto causado pelo risco.

- **Transferir** - A transferência de riscos exige a passagem do impacto negativo de uma ameaça para terceiros, juntamente com a propriedade da resposta. Essa transferência de riscos simplesmente confere a outra parte à responsabilidade por seu gerenciamento; ela não elimina os riscos [PMI 2004]. Essa estratégia é mais eficaz quando se trabalha com transações financeiras. **Transferir riscos geralmente você paga a terceiros um valor para que este assuma seus riscos, inclui ainda seguro, desempenho comprovados, garantias e comprovação.** Os contratos podem ser utilizados para transferir responsabilidades por riscos especificados **para uma outra parte. Em muitos casos, o uso de um contrato com base no custo pode transferir o risco do custo para o comprador, enquanto um contrato de preço fixo pode transferir o risco para o fornecedor.**
- **Mitigar** – Nos projetos de software, a estratégia de mitigação tem se destacado como uma prática **prática adequada, pois ela abrange um universo de modelos(templates) e podem ainda contar com experiências adquiridas em projetos anteriores.** Esta estratégia orienta que se deve **deverem** mensurar as probabilidades da ocorrência dos eventuais riscos e a redução das conseqüências adversas. Para tanto, é importante que o Gerente de Projetos possa identificar os riscos associados aos projetos desde a sua fase inicial [PMI 2003].

Comment [T345]: ???

Comment [T346]: Quem?

Comment [T347]: Como assim?

Formatted: Font: Italic

12.7.2. Estratégia para Riscos Positivos ou Oportunidades

Assim como as estratégias para riscos negativos, as estratégias para riscos positivos ou oportunidades oferecem também três abordagens para tratar os riscos possivelmente positivos às metas do projeto, são elas: Explorar, Compartilhar ou Melhorar.

- **Explorar** – “Esta estratégia pode ser selecionada para riscos com impactos positivos nos pontos em que a organização deseja garantir que a oportunidade seja concretizada. Esta estratégia tenta eliminar a incerteza associada a um risco positivo específico fazendo com que a oportunidade definitivamente aconteça. A exploração de forma direta das respostas inclui a designação de recursos mais capacitados para o projeto a fim de reduzir o tempo para término ou a fim de fornecer uma qualidade maior do que a originalmente planejada” [FERRARI 2004].
- **Compartilhar** – O compartilhamento do risco envolve a atividades de terceiros, ou seja, o risco é atribuído a terceiros **cujao a capacidade de aproveitar melhor a oportunidade em beneficio beneficio do projeto.** Algumas ações compartilhadas são sugeridas como

Comment [T348]: Está faltando algo nessa frase para fazer sentido.

parcerias, equipes, empresas de propósito específico ou *joint ventures* para compartilhamento de riscos que tenham como fundamento claro na a capacidade de gerenciar oportunidades.

Formatted: Font: Italic

Formatted: Font: Italic

- **Melhorar** - Esta estratégia tem como objetivo modificar o "tamanho" de uma oportunidade através do aumento da probabilidade e/ou dos impactos positivos e pela identificação e maximização dos principais acionadores desses riscos de impacto positivo [PMBOK 2004].

12.7.3. Estratégia para Ameaças e Oportunidades

Comment [T349]: Como assim? Aceitação é o que aqui? Só tem aceitação? Não tem nenhuma definição desse termo?

- **Aceitação** - É uma estratégia bastante adotada, pois dificilmente você consegue eliminar todos os riscos de um projeto. A estratégia indica que o plano de gerenciamento do projeto não foi alterado pela equipe para tratar um risco ou que não consegue identificar qualquer outra estratégia de resposta adequada. Essa estratégia pode ser utilizada tanto para as ameaças ou para as oportunidades dos riscos, pode ser ainda ativa ou passiva. A aceitação ativa pode incluir desenvolver um plano de contingência para executar quando ocorrer um risco. A aceitação passiva não requer ação, deixando a equipe de projeto fazer um arranjo quando o risco ocorrer.

Comment [T350]: Tem referências?

12.7.4. Estratégia para Respostas Contingenciais

Certas respostas são projetadas para uso exclusivo de determinados eventos adversos (riscos), principalmente aqueles que surgem no decorrer do projeto. Desenvolvendo um plano de contingência antes se pode reduzir enormemente o custo de uma ação quando ocorrer o risco.

12.8. Monitoramento e Controle de Riscos

“Monitoramento e controle do risco é o processo de identificar e de assegurar o controle do risco, monitorando riscos residuais e identificando novos riscos, assegurando a execução dos planos do risco e avaliando sua eficiência na redução dos riscos.” [PMBOK 2004].

Formatted: Indent: Left: 0", First line: 0"

Este processo de monitoramento e controle de riscos armazena as métricas que se relaciona com os planos de contingência. Este processo é contínuo para todo o ciclo de vida do projeto. Os riscos se

alteram no andamento do projeto, com isso desenvolvem novos riscos ou antecipam a futuros eventos adversos.

Bons processos de monitoramento e controle de riscos, provêem informações que lidam com decisões eficazes o que fazer no progresso de eventos dos riscos.

Segundo o PMI, as propostas para monitoramento e controle de riscos, são para determinar:

- As respostas ao risco estão sendo implementadas como planejadas.
- Ações de respostas ao risco estão eficazes como esperadas ou se novas respostas devem ser desenvolvidas.
- As hipóteses ainda são válidas.
- Análise de tendências da exposição do risco tem mudado prioridades.
- Ocorreu um detonador do risco.
- As políticas e procedimentos adequados estão sendo seguidos.
- Têm ocorrido ou surgido riscos que não foram identificados anteriormente.

Controle de risco pode envolver escolha de alternativas estratégicas, implementando um plano de contingência, tomando ações corretivas ou replanejando o projeto [PMI 2007]. Mostraremos a seguir as principais Técnicas e Ferramentas para o Monitoramento e Controle do Risco;

12.8.1. Auditorias da resposta ao risco do projeto

Os auditores de riscos analisam e documentam o efeito da resposta ao risco em evitar, transferir ou mitigar a presença de eventos adversos ao projeto, além da eficácia do owner do risco. Auditoria do risco é uma técnica empregada durante o processo de monitoramento e controle de riscos, durante o ciclo de controle de risco do projeto.

12.8.2. Revisões periódicas do risco do projeto

Essa técnica deve ser regulamente cronogramada. O risco do projeto é um ponto que deve ser levantado em todas as reuniões da equipe do projeto. Classificação e priorização do risco podem sofrer alterações durante o ciclo de vida do projeto. “Algumas mudanças podem requerer análises de qualificação e quantificação adicionais.”[PMBOK 2007].

Comment [T351]: Coloca a referência aqui.

Comment [T352]: Tem referência?

Comment [T353]: Não entendi esse EM EVITAR aqui.

Formatted: Indent: First line: 0"

Formatted: Font: Italic

Comment [T354]: Acho que essa palavra, por não existir deve estar entre aspas.

Formatted: Indent: First line: 0"

12.8.3. Análise do trabalho realizado

O trabalho realizado é usado para monitorar todo o desempenho do projeto versus um plano inicial (*baseline*). Resultados de uma análise do trabalho realizado podem indicar desvio potencial de custo para concluir o projeto e os objetivos do cronograma [FERRARI 2007]. Quando um projeto começa a desviar de seu planejamento inicial, devem ser realizadas novas análises [análises](#) e atualizações dos riscos.

Formatted: Indent: First line: 0"

Formatted: Font: Italic

12.8.4. Técnica de medição do desempenho

A medição do desempenho técnico observa as verificações técnicas durante a execução do projeto com o cronograma do plano de gerenciamento do projeto de realizações técnicas [FERRARI 2004]. Podemos citar um exemplo de desvio do projeto, quando temos planejado varias [várias](#) funcionalidades para serem entregues, mais no dia da entrega somente algumas foram entregues ao cliente. Essa forma pode prever o grau de [êxodo](#) da realização do escopo do projeto.

Formatted: Indent: First line: 0"

Comment [T355]: É êxodo mesmo?

12.8.5. Planejamento adicional de resposta ao risco

Se um risco emergente que não havia sido previsto no plano de resposta ao risco ou o impacto dele nos objetivos é maior que o esperado, a resposta planejada pode não ser adequada. Será necessário realizar um planejamento adicional para controlar o risco [PMI 2004].

Formatted: Indent: First line: 0"

12.9. Tópicos de Pesquisa

Comment [T356]: Colocar mais tópicos de pesquisa.

1 – Gerenciamento de Riscos Corporativos

Comment [T357]: É importante colocar a referência desse tópico de pesquisa e adicioná-la na seção de referencias.

O gerenciamento de riscos corporativos auxilia os gerentes a se concentrarem-se nos riscos positivos ou negativos, sobre a capacidade de atingir os objetivos estratégicos e agregar valores para os acionistas. Existem vários aspectos que devem ser considerados para estabelecer os objetivos e métricas para o gerenciamento de riscos.

Comment [T358]: Não encaixou bem essa palavra aqui

O gerenciamento de riscos, exerce um papel fundamental no desenvolvimento de projetos de software, . Através de seu planejamento, ele pode mostrar o grau de sucesso de um projeto para uma organização. Nessa forma foram adotados vários modelos, e ferramentas que são utilizados pelas organizações para analisarem os planos estratégicos das empresas, quais os riscos identificados podem comprometer este plano.

Comment [T359]: Seria melhor: e identificarem quais os riscos poderiam comprometer este plano.

12.9.1. Sugestões de Leitura

- Para aquisição de novos conhecimentos relacionados a à gestão de riscos, sugere-se a leitura do PMBOK, ele e book que conta com um conjunto de conhecimentos relacionada relacionado a à gestão de riscos, além além de outras áreas relacionadas a à gestão de riscos como: gestão de projeto de software, custos de um projeto, dentre outros.
- Uma outra Outra fonte de conhecimentos relacionada a riscos, à flexibilidade dos projetos, e a à importância do prazo, temos um blog bem interessante que pode ser acessado pelo link: <http://www.virtue.com.br/blog/?cat=3>, de fácil entendimento de a todos os leitores.
- Um livro bem interessante para a gestão de riscos empresariais que também esta está relacionado ao fator de sucesso nos projetos de software, pode ser recomendado, . ele Ele tem como titulo título Gestão de Riscos Empresariais, tendo como autor Paulo Sergio Sérgio Monteiro Dos Santos, . este Este livro tem 110 páginas.

Comment [T360]: Não seria melhor de OUTROS

Comment [T361]: ?????

Comment [T362]: Deve ser impessoal e esse verbo não cabe aqui. Poderia ser: trata-se de um blog

12.10. Exercícios

1 – Das técnicas que de coleta de informações utilizadas pela gerencia gerência de riscos, na sua opinião, qual você julga as mais importantes aos riscos de um projeto?

Formatted: Font: Not Bold

2 – Na sua opinião, qual a importância da gerencia gerência de riscos para um projeto de software?

3 – Quais os fatores que podem afetar os riscos em um projeto? Defina cada um dos fatores deles?

4 – Sobre as estimativas dos riscos, alguns gerentes acham ideal que os riscos ocorraerem o quanto antes, já possui enquanto outros que alegam que, quanto mais retardar a chegada dos riscos, é melhor para o projeto, e, c Com suas palavras, dê sua opinião sobre esses comentários?

Comment [T363]: Seria melhor: Enquanto

Comment [T364]: Tira esse QUE

5 – Qual a diferença entre análise quantitativa e análise qualitativa dos riscos?

6 – Quais as técnicas com diagramas [que](#) podem ser utilizadas no gerenciamento de riscos? Explique cada uma.

7 – Quais os principais processos utilizados pela gestão de riscos? Explique a importância de cada um individualmente.

8 – Como podem ser classificados os riscos?

9 – Quais os principais requisitos fundamentais que devem ser abordados pelo planejamento do gerenciamento de riscos.

10 – Qual a importância da matriz de probabilidade e impacto para um projeto de software?

12.11. Referências

Comment [T365]: Rever referência uma a uma, pois praticamente todas estão fora do padrão do SBC.

http://www.pming.org.br/downloads/GestaoRiscosProjetos_LucioDiniz_31082004.pdf
<http://www.linhadecodigo.com.br/livros.asp?id=619>
http://www.devmedia.com.br/articles/viewcomp.asp?comp=10585&hl=*riscos*
http://www.devmedia.com.br/articles/viewcomp.asp?comp=9529&hl=*riscos*
http://www.bfpug.com.br/islrig-rio/Downloads/Gerencia_de_Riscos.pdf
<http://www.devmedia.com.br/articles/viewcomp.asp?comp=8028>
http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_11.4.html
http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_11.2.html
http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_11.3.html
http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_11.4.html
http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_11.5.html
http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_11.6.html
<http://wpm.wikidot.com/area:gerenciamento-de-riscos>
<http://wpm.wikidot.com/tecnica:analise-e-reunioes-de-planejamento>
<http://wpm.wikidot.com/artefato:plano-de-gerenciamento-de-riscos>
<http://wpm.wikidot.com/tecnica:revisoes-da-documentacao>
<http://wpm.wikidot.com/tecnica:tecnicas-de-coleta-de-informacoes>
<http://wpm.wikidot.com/tecnica:tecnicas-com-diagramas>

EVANS, JAMES R.; OLSON, DAVID L. Introduction to simulation and risk analysis. New Jersey: Prentice Hall / Upper Saddle River, 1998

SOUZA, Marcelo C. Mesquita de. Quantificação das Incertezas na Avaliação de Projetos: O Modelo Utilizado na Agência de Fomento do Estado da Bahia. 2004. 134f.

Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2004
Lopes, S. e Souto, S. Documento de especificação de projeto versão 1.0. In: Suppera Solutions Relatório Técnico, Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil

Myerson, M. (1996) Risk Management Processes for Software Engineering Models, Artech House

Boehm, B. W. (1989), Tutorial: Software Risk Management, IEEE Computer Society Press.

Heldman, Kim – Gerencia de Projetos – Ed. Campus- 2006

Schmitz, Eber Assis / Alencar, Antonio Juarez - Análise de Risco em Gerencia de Projetos - Brasport – 2006

Pressman, Roger S. Engenharia de software – São Paulo: Makron Books, 1995

Qualidade de Software – Engenharia de Software Magazine

BARBOSA, Igor de Mesquita. Análise de Características de Projetos de Software para a Definição de Processo de Software. Trabalho de conclusão de curso, Universidade de Pernambuco, 2005

PMBOK. A Guide to the project Management Body of Knowledge. PMI Project Management Institute. 3ª edição. PMI-USA. (versão traduzida para português 2004). 69 Disponível em: <<http://www.cos.ufrj.br/~mareas/livros/PMBOK.pdf>>. Acesso em: 11 jun 2009

PRESSMAN, Roger S. Engenharia de Software; tradução Rosângela Dellosso Pentead, revisão técnica Fernão Stella R. Germano, José Carlos Maldonato, Paulo César Masiero. - 6.ed. – São Paulo: Mc Graw – Hill, 2006

TRIGO, Thiago; LINS, Arthur; e, GUSMÃO, Cristine. CBR Risk: Um Modelo para Identificação Automática de Riscos em Projetos de Software. RBGP. Revista Brasileira de Gerenciamento de Projetos, v. 15, p. 11-17, 2007

PMI (Project Management Institute) Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos – Guia PMBOK® 3. ed. Upper Darby, 2004

PMI (Project Management Institute) A Guide to the Project Management Body of Knowledge – Guia PMBOK® 4. ed. Upper Darby, 2008

Vargas, R. (1999). Gerenciamento de Projetos - Estabelecendo diferenciais competitivos, 6ª edição, Editora Braspor

Vargas, R. (2007). Manal prático do plano de projeto: utilizando o PMBOK Guide. 3.ed. Rio de Janeiro: Brasport

Verzuh, E. (2000). MBA compacto, Gestão de Projetos. 3. ed. Rio de Janeiro: Campus

SUMÁRIO

13. Gestão de Riscos Em Projetos de Software	01
13.1. Gestão de Riscos	01
13.2. Análise e Reuniões de Planejamento de Riscos	03
13.3. Planejamento do Gerenciamento de Riscos	04

13.4.	Identificação dos Riscos.....	07
12.4.1.	Classificação dos Riscos	08
12.4.2.	Métodos para Coleta de Dados para Identificação dos Riscos.....	08
12.4.2.1.	Revisões da documentação.....	09
12.4.2.2.	Técnicas de Coleta de Informações.....	09
12.4.2.3.	Análise da Lista de Verificação	10
12.4.2.4.	Análise das Premissas	10
12.4.2.5.	Técnicas com Diagramas	10
13.5.	Análise Qualitativa de Riscos	12
12.5.1.	Avaliação de Probabilidade e Impacto dos Riscos.....	13
12.5.2.	Matriz de Probabilidade e Impacto	13
12.5.3.	Avaliação da Qualidade dos Dados Sobre os Riscos	15
12.5.4.	Categorização dos riscos	15
12.5.5.	Avaliação da Urgência do Risco	15
13.6.	Análise Quantitativa de Riscos	16
12.6.1.	Técnicas de Representação e Coletas de Dados.....	16
12.6.2.	Análise quantitativa de riscos e técnicas de modelagem.....	18
13.7.	Planejamento de Respostas a Riscos.....	19
12.7.1.	Estratégia para Riscos Negativos ou Ameaças	20
12.7.2.	Estratégia para Riscos Positivos ou Oportunidades	21
12.7.3.	Estratégia para Ameaças e Oportunidades	22
12.7.4.	Estratégia para Respostas Contingenciais	22
13.8.	Monitoramento e Controle de Riscos.....	22
12.8.1.	Auditorias da resposta ao risco do projeto	23
12.8.2.	Revisões periódicas do risco do projeto.....	24
12.8.3.	Análise do trabalho realizado	24
12.8.4.	Técnica de medição do desempenho	24
12.8.5.	Planejamento adicional de resposta ao risco.....	24
13.9.	Tópicos de Pesquisa	25
12.9.1.	Sugestões de Leitura	25
13.10.	Exercícios.....	26
13.11.	Referências	29

Sumário

GESTÃO DE PESSOAS 262

14.1 CONCEITOS E DESAFIOS 263

14.1.1	DESAFIOS ORGANIZACIONAIS PARA O GERENCIAMENTO DE PESSOAS.	263
14.1.1.1	AVANÇOS TECNOLÓGICOS	263
14.1.1.2	<i> Downsizing</i>	264
14.1.1.3	CULTURA ORGANIZACIONAL.....	264
14.1.2	DESAFIOS INDIVIDUAIS	265
14.1.2.1	IDENTIFICAÇÃO COM A EMPRESA.....	265
14.1.2.2	CONDUTA ÉTICA.....	265
14.1.2.3	<i> Empowerment</i>	265

14.2 MOTIVAÇÃO: CONCEITOS E TEORIAS 266

14.2.2	TEORIAS DE MOTIVAÇÃO.....	267
14.2.3	PROCESSOS DE MOTIVAÇÃO.....	268

14.3 TRABALHO EM EQUIPE 270

<FALTA COMPLEMENTAR> ERROR! BOOKMARK NOT DEFINED.

14.3.3	TIPOS DE EQUIPE.....	270
--------	----------------------	-----

14.4 GESTÃO DE PESSOAS POR COMPETÊNCIAS 271

14.4.2	REMUNERAÇÃO ESTRATÉGICA	273
14.4.3	AVALIAÇÃO DE DESEMPENHO.....	274

14.5 LIDERANÇA 275

14.5.1	O PAPEL DO LÍDER.....	275
14.5.2	<i> Coach</i>	276

14.6 GERENCIAMENTO DE CONFLITOS 277

14.6.1	VISÃO DOS CONFLITOS.....	277
14.6.2	NÍVEIS DE CONFLITO.....	277
14.6.3	CONFLITO E ESTRESSE	278

<u>14.7</u>	<u>GESTÃO DE PESSOAS E DESENVOLVIMENTO DA INTELIGÊNCIA EMOCIONAL</u>	<u>279</u>
14.7.1	CONHECIMENTOS, HABILIDADES E ATITUDES (CHA)	280
<u>14.8</u>	<u>CONSIDERAÇÕES FINAIS</u>	<u>281</u>
<u>14.9</u>	<u>SUGESTÕES DE LEITURA</u>	<u>282</u>
<u>14.10</u>	<u>TÓPICOS DE PESQUISA</u>	<u>284</u>
<u>14.11</u>	<u>EXERCÍCIOS</u>	<u>285</u>
<u>REFERÊNCIAS</u>		<u>286</u>

14

Gestão de Pessoas

Graziela Simone Tonin

Gerenciar Projetos há alguns anos, já deixou de ser uma arte. Inúmeras metodologias ensinam passo a passo como atingir metas e decompor um projeto, podendo assim chegar a uma grande riqueza de detalhes, basta seguir uma dessas metodologias. Porém, na grande maioria dos casos tem se limitado a isso, a seguir uma metodologia, esquecendo que empresas são organismos vivos, altamente complexos em razão da interação entre pessoas, ferramentas, procedimentos, comunicação, políticas, cultura, tecnologia, clientes e entre tantos outros fatores. Ou seja, planejar e executar um projeto e um empreendimento humano, um empreendimento guiado por pessoas. E pessoa tem sentimentos, desejos, expectativas, conhecimentos e conflitos que influenciam e muito a realização das metas de um projeto. Definir o escopo do projeto, determinar prazos e controlar os custos é algumas das muitas atividades do projeto, porém não há nada mais complexo no escopo de um projeto que gerenciar e tratar as expectativas de todos os envolvidos na sua execução, especialmente, do time do projeto. Por isso, este capítulo visa estudar uma das mais importantes e difíceis áreas de conhecimento que um gerente de projetos deve possuir para conseguir o sucesso de seu empreendimento: a gestão de pessoas. Pessoas são capazes de conduzir a empresa a seus objetivos estratégicos, gerando mudanças, riqueza e valor. Os seres humanos são a chave do sucesso ou do fracasso das estratégias inovadoras de uma organização. Se os processos e a tecnologia são necessários para transformar uma empresa, é importante também lembrar que são as pessoas que os conduzem, ou seja, sem elas de nada adiantariam processos ou tecnologias. No primeiro capítulo será abordado sobre conceitos e os grandes desafios, hoje ainda encontrados na Gestão de Pessoas. Logo após será percorrido no segundo capítulo, sobre motivação sua importância, teorias e processos existentes, que foram desenvolvidas com o intuito de auxiliar a este fator importantíssimo que influencia diretamente no desempenho das pessoas. Já no terceiro capítulo será abordado sobre trabalho em equipe. No quarto capítulo será comentado sobre uma importante área que vem ganhando ênfase nos últimos tempos, a gestão de pessoas por competências. Em seguida no capítulo cinco será estudado sobre um dos fatores de maior influência na gestão de pessoas, a liderança. No capítulo seis será abordado sobre um dos fatores considerados mais críticos na vida de um gerente de pessoas, a gestão de conflitos. E no último capítulo será percorrido sobre inteligência emocional e onde esta influencia, onde que usada pode ser um diferencial para se ter uma equipe eficaz. E por fim, algumas considerações finais sobre todos os assuntos abordados neste capítulo. Sugestões de boas leituras para quem quiser se aprofundar no assunto. Alguns tópicos de pesquisa que ficam como sugestão para trabalhos

a lista de referencias utilizada onde também, podem ser encontradas matérias mais aprofundadas, sobre os diversos fatores e desafios na área de gestão de pessoas.

14.1 Conceitos e Desafios

Segundo Laércio Cosentino, presidente da Microsiga, o RH 'é peça fundamental para qualquer organização. "Repito sempre, que quando as empresas têm a mesma tecnologia, a diferença são as pessoas. Com esse diferencial em mente, cabe ao RH alinhar o negócio com as pessoas", explica. Dessa maneira, o profissional da área deve entender as necessidades de um departamento e identificar o fator crítico de sucesso para ele. "Para que isso aconteça, é necessário que ele trabalhe em conjunto com outras áreas", diz Cosentino.

Segundo o PMI Um projeto 'é um empreendimento único, com inicio e fim bem definidos, que utiliza recursos limitados e conduzido por pessoas visando atingir metas e objetivos pré-definidos, estabelecidos dentro de parâmetros, prazo, custo e qualidade. [PMI 2004]

E para ser executado um projeto precisa ser gerenciado e isto 'é feito por pessoas. Segundo Koontze e O' Donnel, gerenciar consiste em executar atividades e tarefas que tem com propósito planejar e controlar atividades de outras pessoas para atingir objetivos que não podem ser alcançados caso as pessoas atuem por conta própria, sem o esforço sincronizado dos subordinados. [Koontze1980]

Segundo Fisher e Fleury, gestão de pessoas 'é o "Conjunto de políticas e práticas definidas de uma organização para orientar o comportamento humano e as relações "Interpessoais no ambiente de trabalho." [Fisher e Fleury 1992]. Em meados dos anos 80, o número de diretorias de RH era pequeno, conforme explica Dárcio Crespi, presidente da Heidrick & Struggles. "Havia o RH sem expressão, pendurado em outros departamentos, subordinado à área administrativa ou financeira, o que gerava comentários sobre a competência desses profissionais." Isso quer dizer que o RH era o reduto de todos os que não encontravam sucesso em outras áreas. Com o processo de globalização a partir dos anos 90, o mundo corporativo despertou para a importância das pessoas. [Revista Melhor 2008]

E isso não foi diferente nas empresas de TI, pois, essas mais do que qualquer outra empresa precisam possuir talentos e desenvolve-los, logo precisam de um RH estratégico.

Abaixo será **discorrido** sobre os principais desafios ainda existentes na Gestão de Pessoas.

14.1.1 **Desafios** Organizacionais para o Gerenciamento de Pessoas.

A competitividade entre as empresas esta cada vez maior e em nível mundial. Os desafios para os gerentes neste contexto de competição aumentam a cada dia mais. Abaixo será discorrido sobre alguns fatores importantes e que cada vez mais precisam ser observados pelos **gestores**.

Comment [LA366]: Rever se a introdução deve ser escrito em italico

Comment [LA367]: Muitos leitores podem nao entender o significado da palavra, talvez descrito poderia ficar melhor.

Comment [LA368]: A seção deve esta alinha ao paragrafo

Comment [LA369]: Reveja o espaçamento entre paragrafos de acordo com o template.

O ambiente em que as empresas se situam, estão constantemente mutáveis. Novas tecnologias surgem a todo instante, assim, logo após a introdução de uma tecnologia, surge à necessidade de inovar. E esta tem a necessidade de adotar novas tecnologias, pois, se quiserem prosperar, ou às vezes, pelo menos sobreviver deverão ajustar-se rápido e efetivamente as mudanças. E para trabalhar com as mudanças, precisarão de trabalhadores cada vez mais qualificados. Segundo Gil, nesse contexto, o capital humano passa a ser entendido como estratégia fundamental para enfrentar transformações radicais. Logo, a capacidade de gerar novos conhecimentos e agregá-los ao processo produtivo e que se torna a grande vantagem competitiva. [GIL, 2008] Portanto, trabalhador passa ser a diferença, e além de dedicação pessoal cabe a gestores proporcionar meios necessários para garantir este ajustamento.

Comment [LA370]: Sem virgule entre o autor e ano.

14.1.1.2 Downsizing

Segundo o economista Paulo Nunes, *downsizing*, surgiu como uma necessidade de solucionar o excesso de burocracia que se verificava em muitas organizações, o que dificultava a tomada de decisões e a adaptação às novas realidades ambientais. Resultando, assim na redução dos níveis da gestão e na redução da dimensão da organização através da anulação de áreas produtivas não essenciais, centrando-se no que melhor sabem fazer (*core* competência), subcontratando ao exterior (*outsourcing*) atividades não fundamentais para o *core-business* permitindo uma maior flexibilização da estrutura organizacional.

Comment [LA371]: O estilo desse parágrafo esta diferente dos demais. Tente adotar um padrão.

Apesar das vantagens dos cortes nos custos e na maior flexibilização da organização, normalmente através da anulação de atividades relacionadas com áreas ou produtos menos rentáveis, o que permite uma concentração de esforços e recursos nas áreas mais rentáveis, o *downsizing* continua a ser muito criticado e mal aceite devido ao problema dos despedimentos. Pois, na aplicação prática do *downsizing* consistiu, na maioria das vezes, no corte de custos em áreas consideradas não essenciais, resultando geralmente no despedimento de numerosos trabalhadores. Por isso, muitos gurus da gestão criticaram ferozmente a utilização do *downsizing* pelas organizações. Nomes como Peter Drucker e Charles Handy viriam a público lançar a sua indignação. Robert Tomasko, um dos autores do conceito, afirmou mesmo que se teria ido longe de mais na utilização do *downsizing*. Michael Hammer, autor do *Best seller* “*Beyond Reengineering*”, explica que a reengenharia não tinha nada a ver com o *downsizing* e que deveria ser aplicada para o crescimento das empresas, não para o corte de custos. Prahalad, que escreveu com Gary Hamel o livro “*Competing for the Future*”, chamou ao *downsizing* de “anorexia empresarial”.

14.1.1.3 Cultura Organizacional

A cultura organizacional ou cultura corporativa é o conjunto de hábitos e crenças estabelecidos através de normas, valores, atitudes expectativas compartilhados por todos os membros da organização. Refere-se ao sistema de significados compartilhados por todos os membros e que distingue uma organização das demais. Constitui o modo institucionalizado de pensar e agir que existe em uma organização.

A essência da cultura de uma empresa é expressa pela maneira como ela faz seus negócios, a maneira como ela trata seus clientes e funcionários, o grau de autonomia ou

seus funcionários com relação à empresa. A cultura organizacional representa as percepções dos dirigentes e funcionários da organização e reflete a mentalidade que predomina na organização. Por esta razão, ela condiciona a administração das pessoas. Influenciando diretamente a motivação, o desempenho humano e a satisfação no trabalho. Surgindo assim muitos desafios, para que se possa ter uma relação recíproca entre a cultura da empresa e a cultura pessoal de cada funcionário a qual norteia suas atitudes, e sendo um dos atores coadjuvantes nessa relação, o gerente. Pois, esse exerce grande influencia no clima organizacional.

14.1.2 Desafios Individuais

Os desafios individuais referem-se à postura adotada pela empresa em relação aos seus empregados. E isso começa já na abordagem das competências de todos os indivíduos, que trabalham na organização, para identificar o conjunto de competências que seus funcionários precisam aplicar para sustentar a competência essencial da empresa. E no grau de investimento despendido pela empresa aos seus funcionários.

14.1.2.1 Identificação com a Empresa

Segundo Gil, os empregados mais produtivos são os que mais se identificam com a cultura da empresa e com seus objetivos são os verdadeiros parceiros da empresa. [GIL 2009]. Por isso, a empresa necessita desenvolver estratégias para atrair e manter empregados com essas características. Isso requer muita transparência nas empresas no estabelecimento das bases de sua política de pessoal. E no lado do funcionário também requer que uma mesma transparência que somente poderá ser adota com uma conduta-ética

14.1.2.2 Conduta Ética

Existem muitos desafios de dilemas-éticos nas organizações, pois, a despeito da forma positiva com que as organizações procuram tratar as questões éticas, acaba-se gerando muitas vezes as condutas antiéticas. Problemas como, conflitos de interesse, problemas ambientais, comparação de renda de funcionários, assédio moral são constantes nas empresas. E isso está diretamente ligado ao valor que é dispensado ao funcionário.

Segundo o médico Dráuzio Varela, “_os olhos da sociedade, a mera existência de um juramento solene da à impressão de que somos sacerdotes e de que devemos dedicação total aos que nos procuram, sem manifestarmos preocupação com os aspectos materiais como as condições de trabalho ou remuneração pelos serviços prestados, para a felicidade de tantos empresários gananciosos” [VARELLA 2006].

Portanto, para que se tenha um saudável relacionamento empresa funcionário e funcionário empresa e muito importante que cada empresa tenha um código de ética, que seja explanado para seu funcionário assim que este inicia na empresa, onde ambos usem seu bom senso para alcançarem seus objetivos.

14.1.2.3 Empowerment

Empowerment significa a descentralização de poderes pelos vários níveis hierárquicos da empresa, o que se traduz em incentivos para a tomada de iniciativas em benefício da empresa.

A tabela abaixo destaca as principais diferenças de uma empresa com e sem *empowerment*. [EXECUTIVE DIGEST 1996]

Tabela 14.1.2.3 Diferenças de uma empresa com e sem *Empowerment*.

SEM	COM
Esperar ordens	Tomar decisões
Executar	Planejar e Executar
Chefe Responsável	Todos responsáveis
Encontrar Culpados	Resolver Problemas

14.2 Motivação: Conceitos e Teorias

Tanto motivação como quanto emoção vem do verbo latino *movere*, que significa mover-se. Ambas indicam um estado de despertar do organismo. Logo, motivação é a força que estimula as pessoas a agir. No passado acreditava-se que esta força era determinada principalmente pela ação de outras pessoas. Hoje, sabe-se que a motivação tem origem numa necessidade. Assim, cada um de nós dispõe de motivações próprias geradas por necessidades distintas e não se pode, a rigor, afirmar que uma pessoa seja capaz de motivar outra. Motivação é consequência de necessidades satisfeitas. Essas necessidades são intrínsecas as pessoas. Isso significa que os gerentes não são capazes de motivar, mas de satisfazer às necessidades humanas ou contra fazê-las. [ARCHER 1990]

Todo comportamento humano é motivado. Não devemos confundir estímulo com motivação: no primeiro caso, não há envolvimento do eu interior. No incentivo, o indivíduo age levado por pressões externas: ganhar comissões, evitar punições. Na motivação ao contrário, o eu interior está envolvido e a pessoa age impelida por uma força interna, porque gosta e quer.

O estímulo faz com que as pessoas ajam sob condições e tempos limitados. Uma condição duradoura, entretanto, só pode emanar de uma motivação verdadeira, que ocorrerá quando o indivíduo tiver seu próprio gerador instalado dentro de si, não havendo necessidade de impulsos externos e terá vontade de executar as tarefas.

14.2.1 A Importância da Motivação

O mundo cada vez mais competitivo dos negócios exige altos níveis de motivação das pessoas. Empregados motivados para realizar seu trabalho, tanto individualmente como em grupo, tendem a proporcionar melhores resultados. A maioria dos teóricos da

A motivação é a chave do comprometimento dos colaboradores. É muito mais fácil para as empresas conseguirem pessoas competentes do que comprometidas. Por isso, identificar fatores capazes de promover a motivação dos empregados e dominar as técnicas adequadas para trabalhar com ela vem se tornando tarefa das mais importantes para os gerentes.

Comment [LA372]: Reveja o padrão de estilo do seu texto.

14.2.2 Teorias de Motivação

Existem muitas teorias que tratam desde fatores higiênicos e motivacionais como a Teoria de *Herzberg*: onde a ausência destes fatores como benefícios, remuneração justas, e ambiente adequado de trabalho, conteúdo do cargo de trabalho, ligados a desafios e oportunidades, estima e realização, podem causar muita insatisfação; *Teoria das Necessidades de Alderfer*: esta é similar a de Maslow e esta divide em três níveis, o primeiro refere-se a nossas preocupações, o segundo refere-se à motivação para mantermos relacionamentos interpessoais e o terceiro refere-se ao desejo intrínseco do indivíduo para o desenvolvimento pessoal; *Teoria das Necessidades de Conquistas Pessoais de McClelland*: onde o principal aspecto desta teoria é que os indivíduos são motivados para evitar falhas (associado frequentemente com objetivos de desempenho) ou lançar sucesso (frequentemente associado com objetivo de poder). É a teoria que será discutida com mais detalhes abaixo, por ser uma das teorias mais explorada e aplicada ao longo dos anos.

Comment [LA373]: Estilo de formatação do seu texto

Teoria Humanística da Hierarquia das Necessidades de Maslow:

Está dividida em cinco níveis que são explicados abaixo e representados pela figura.

Comment [LA374]: Como não é uma seção, não há necessidade desse nome esta numa fonte maior, pode continuar em seguida, comente em negrito.

- **Necessidades Fisiológicas:** São relacionadas às necessidades do organismo, e é a principal prioridade do ser humano. Entre elas estão respirar e se alimentar. Sem estas necessidades supridas, as pessoas sentirão dor e desconforto e ficarão doentes.
- **Necessidades de Segurança:** Envolve a estabilidade básica que o ser humano deseja ter. Por exemplo, segurança física (contra a violência), segurança de recursos financeiros, segurança da família e de saúde.
- **Necessidades Sociais:** Com as duas primeiras categorias supridas, passa-se a ter necessidades relacionadas à atividade social, como amizades, aceitação social, suporte familiar e amor.
- **Necessidades de Status e Estima:** Todos gostam de ser respeitados e bem vistos. Este é o passo seguinte na hierarquia de necessidades: ser reconhecido como uma pessoa competente e respeitada. Em alguns casos leva a exageros como arrogância e complexo de superioridade.
- **Necessidade de Auto Realização:** É uma necessidade instintiva do ser humano. Todos gostam de sentir que estão fazendo o melhor com suas habilidades e superando desafios. As pessoas neste nível de necessidades

de ajudar aos outros. Suprir esta necessidade equivale a atingir o mais alto potencial da pessoa.

Figura 14.2.1- Hierarquia das Necessidades de Abraham Maslow



Fonte: Fatores Motivacionais Da Comunidade Científica Para Publicação E Divulgação De Sua Produção Em Revistas Científicas

Comment [LA375]: Não tenho certeza, mas acho que as figures devem ter fontes de onde foram extraídas (referências bibliográficas)
- Fonte de figuras deve ser Helvética 10

14.2.3 Processos de Motivação

Existem muitas teorias que tentam descrever como as pessoas são motivadas. Na tabela abaixo será feita uma breve citação sobre cada uma delas:

Tabela 14.2.3 - Teorias dos Processos de Motivação

Nome da Teoria	Descrição
<i>Teoria X de McGregor</i>	<ul style="list-style-type: none"> • Esta teoria foi desenvolvida para descrever a relação entre os gerentes e subordinados. • Nela o homem é apresentado com um ser carente, que não gosta de seu trabalho e se limita a fazer o necessário, para subsistir, não se importando com a realização pessoal. • Onde as pessoas são preguiçosas, não gostam de trabalhar, precisam de recompensas e punições para agir. • São imaturas e incapazes de assumir responsabilidades.

<p><i>Teoria Y de McGregor</i></p>	<ul style="list-style-type: none"> • Essa teoria tem ótima base à integração, por ela assegurar e validar a autoridade. • Sugere autocontrole quando as necessidades dos projetos e membros da equipe são reconhecidas. • Segundo esta teoria as pessoas tem necessidades psicológicas de trabalhar e realizar-se profissionalmente. • São maduras e gostam de assumir responsabilidades, podendo conduzir seu próprio trabalho sem supervisão e querem alcançar seus objetivos no projeto.
<p><i>Teoria Cognitiva da Expectativa de Vroom</i></p>	<ul style="list-style-type: none"> • As pessoas pensam cuidadosamente na quantidade de esforço que deveriam dedicar a uma tarefa antes de realizá-la. A motivação aparece se há uma expectativa de um desfecho favorável. • E baseada no conceito de que as pessoas escolhem comportamentos que acreditam que irão conduzi-los a recompensas ou promoções desejadas isto é, o desejo da recompensa e forte o bastante para fazer o esforço valer à pena.
<p><i>Teoria Cognitiva do Reforço de Skinner</i></p>	<ul style="list-style-type: none"> • Baseada no conceito de como as pessoas aprende. • Enfatiza que se um comportamento desejável será repetido se for recompensado e um comportamento indesejado pode ser desencorajado por punição. • Tem ótimos elementos chave quatro técnicas básicas de reforço,

	<p>punição e extinção.</p> <ul style="list-style-type: none"> • Segundo elas os membros da equipe que tiverem comportamentos que tragam resultados, utilizando reforços positivos sua extinção devem ser estimulados; • E os membros que trazem problemas com reforços negativos ou punições devem ser desestimulados.
--	--

Essas teorias enfatizam a importância da motivação e ajudam a compreender melhor as atitudes dos funcionários e como se deve reagir a tais.

Comment [LA376]: Organizar melhor esse texto.

14.3 Trabalho em Equipe

A formação de uma equipe de projetos não se dá somente pela união das pessoas visando à realização de uma dada tarefa. Para se construir uma relação firme é preciso à discussão de valores, visão, missão, expectativas e normas segundo as quais a equipe irá operar em um determinado projeto. Isso deve ser feito antes de se aproximar da definição do trabalho.

14.3.1 Equipes de Projetos Tradicionais Versus Equipes de Auto Desempenho

Na equipe de projetos tradicionais em projetos, o aprendizado contínuo e pouco recompensado. A pessoa tem dificuldade para ver como contribuem para o produto ou serviço final e nunca se envolvem na resolução de problemas. Os gerentes atribuem tarefas, analisam o desempenho e decidem quais serão os procedimentos de trabalho sem a contribuição dos funcionários. Na remuneração, normalmente todos recebem as mesmas recompensas financeiras, independentemente de seu desempenho. O acesso de informação aos dados e aos sistemas de informação é rigidamente controlado, onde estas ficam detidas nas mãos de técnicos e especialistas. A tecnologia é considerada mais importante que as pessoas.

Comment [LA377]: Não há necessidade de iniciar com um recuo, sendo o primeiro parágrafo.

Enquanto que na equipe de projeto de auto-desempenho o contrário acontece, e também na estrutura organizacional existem apenas alguns níveis de gerenciamento entre gerência e subordinados. A organização é muito horizontalizada. No relacionamento com o cliente todos tem um cliente interno ou externo e buscam constantemente entender e suprir as necessidades do cliente. A equipe explora os progressos tecnológicos e busca encontrar formas inovadoras de utilizar a tecnologia existente. Valoriza-se o trabalho em equipe, a participação, a inovação, a qualidade, tanto quanto os lucros. Enfim todos se sentem pessoalmente responsáveis pelo desempenho geral da equipe.

Equipes são alicerces das organizações de alto desempenho. Por mais que tentemos, é impossível chegar ao alto desempenho sem elas. A escolha do tipo certo de equipe não é, porém, tão simples como poderia parecer, uma vez que, conforme mencionado por *Boyett e Boyett* (1999), existe uma gama enorme de opções: equipes de trabalho, equipes interfuncionais, equipes de projeto, equipes de resolução de problemas, equipes autogerenciadas, entre outras. Independentemente dos nomes, o que se observa são três tipos de equipe, que interagem entre si: de trabalho, de melhoria e de integração:

- **Equipes de Trabalho:** projetam, fabricam e oferecem um produto ou serviço a um cliente interno ou externo. São compostas de pessoas que atuam na linha de frente da maioria das organizações, fazendo pesquisas, fabricando produtos, vendendo, prestando serviço aos clientes e realizando a maioria das tarefas que contribuem para os resultados da organização. Incluem-se aí equipes de produção de produtos manufaturados, equipes de desenvolvimento de novos produtos, equipes de propostas, equipes de consultoria, equipes de vendas e serviços, entre outras.
- **Equipes de melhoria:** fazem recomendações de mudança na organização, processos e tecnologia, a fim de se melhorar a qualidade, o custo e o cumprimento dos prazos de entrega dos produtos e serviços. Ao contrário das equipes de trabalho, estas, freqüentemente são temporárias. São criadas para lidar com um problema ou projetos específicos e depois se dissolvem. Equipes de projetos, grupos de auditoria, equipes de qualidade, forças-tarefa, equipes de melhoria do processo e grupos semelhantes para a resolução do problema são exemplos de equipes de melhoria.
- **Equipes de interação:** garantem a coordenação do trabalho em toda a organização. Essas equipes reúnem duas ou mais equipes de trabalho e equipes de melhoria interdependentes em torno de um foco comum, como um cliente, uma linha de produtos, uma tecnologia ou um mercado ou área de serviço geográfico específico. As equipes de integração servem como vínculos entre as equipes de trabalho e as de melhoria. Normalmente, são compostas de membros das equipes interdependentes que estão sendo ligadas. Por exemplo, engenheiros de software das equipes de trabalho de desenvolvimento poderiam fazer parte da equipe de integração formada entre duas equipes de desenvolvimento de software. As equipes gerenciais representam uma forma especial de equipe de integração que define a direção estratégica, estabelece metas e monitora o desempenho das equipes de trabalho e melhoria.

14.4 Gestão de Pessoas por Competências

Na Idade Média, competência era considerada como "capacidade de apreciar e julgar certas questões ou realizar determinados atos."

"O senso comum usa a palavra competência para designar pessoa qualificada para realizar algo".

O setor de recursos humanos passou a testar diversas alternativas ao modelo *Taylorista*

O alinhamento das políticas de gerenciamento das pessoas com as estratégias empresariais levou ao conceito de competências a ser utilizado como base aos novos projetos de gestão de competências para servir de base aos novos processos de remuneração por competências, recrutamento interno por competências, plano de sucessão por competências, desenvolvimento das lideranças por competências e avaliação de desempenho por competências. Pode-se dizer que hoje, além da gestão de competências, que se propõe a elevar em nível de capacitação de colaboradores existe a gestão de pessoas por competências isto é, processos de gerenciamento baseados no mapeamento de competências para tornar esse gerenciamento mais ligado à estratégia da empresa, das unidades ou dos projetos.

A diferença entre gestão de competências e gestão por competências está muito além de uma simples troca de preposições. Gestão de competências é o processo que tem como objetivo adequar às competências certas, no lugar certo, na hora certa, ao custo justo para a organização. Este processo é baseado no mapeamento, desenvolvimento e alocação das competências necessárias ao negócio.

Gestão por competências é a encampação do conceito de competências como base para o gerenciamento de pessoas de uma organização. Isso significa que todos os processos operacionais de gerenciamento de pessoas até a sua remuneração. Os novos processos que os GPs irão aplicar no cotidiano, com o conceito de competências, estão descritos a seguir.

14.4.1 Seleção de Pessoas

Muitas vezes ao longo de projetos é necessário substituir pessoas. Uma forma de realizar o recrutamento é fazendo as análises das competências que serão necessárias para as tarefas que o cargo exige verificando se estas competências existem no candidato. Para isso, tem sido utilizado técnicas de entrevistas por competências, baseada na elaboração previa de perguntas que visam obter relatos das experiências.

Segundo Raj, isto indicara a existência ou ausência das competências que estão sendo pesquisadas e que são necessárias para o bom desempenho das tarefas do profissional que está sendo selecionado.

E hoje a grande dificuldade das organizações no é captar talentos, tem-se um mercado carente de profissionais capacitados. Além disso, grande parte das organizações ao contratar novos colaboradores desconhece o perfil de competências necessárias para o cargo em aberto e o que é pior, muitas organizações não realizam nenhum tipo de processo seletivo baseando-se em indicações de amigos para preencher o cargo. O que termina ocasionando a contratação de um profissional não adequado para a função, causando alta rotatividade de colaboradores e aumentando significativamente os custos dessas organizações.

A Seleção por Competências fundamenta-se em fatos e evidências e não faz uso de inferências nem deduções. Esta metodologia tem como diferencial a objetividade e o foco que outras metodologias de Seleção não apresentam. Visa facilitar a captação de talentos e consiste em identificar no candidato as competências técnicas e comportamentais exigidas pelo cargo possibilitando ao selecionador a percepção de características que nenhum currículo pode fornecer, como planejamento e organização, visão sistêmica, negociação, trabalho em equipe, orientação para resultados etc. Permitindo alinhar o conjunto de competências dos profissionais às necessidades estratégicas da organização.

Para utilizar esta metodologia o primeiro passo é construir o Perfil de Competências:

A construção do perfil de competências (PC) requer a análise e identificação dos indicadores de Competências fundamentais para o cargo. E deve ser realizada com a participação da área requisitante, estabelecendo o início de uma parceria que irá seguir por todo o processo seletivo. É importante frisar que tanto selecionador como requisitante tem papel crucial para o sucesso do processo seletivo.

A área requisitante irá fornecer informações sobre:

- Descrição do cargo atualizada;
- Mudanças ocorridas recentemente;
- Principais desafios do cargo;
- Maiores erros cometidos no cargo;
- Expectativas da área em relação ao cargo;
- Dificuldades e pontos críticos do cargo;
- Principais projetos a serem desenvolvidos.

Através dessas informações serão identificados os indicadores de competências que darão origem ao Perfil de Competências que deverá estar alinhado a Missão, Visão, Valores e Estratégias (MVVE) da organização. E com base neste perfil o profissional de seleção vai estruturar e planejar todo o processo seletivo e desenvolver as ferramentas que irão assegurar objetividade e foco.

Ao utilizar a seleção com foco em competências como metodologia de trabalho o profissional de RH garante a eficácia na escolha do candidato captando talentos em menor tempo e com redução de custos o que é fundamental para obter bons resultados e garantir o êxito nas contratações.

14.4.2 Remuneração Estratégica

A gestão salarial baseada em competência tem sido reconhecida como uma forma de recompensa que favorece o atendimento das novas demandas do mundo competitivo, na medida em que possibilita: ganhos recíprocos; crescimento sustentado; redução da subjetividade; clareza e transparência e senso de propriedade. Ou seja, é uma forma de obter maior vínculo entre as empresas e seus funcionários.

Um conceito que está ligado intimamente com o modelo de remuneração por competência é o de complexidade. As entregas, ou o valor agregado pelas pessoas, devem ser coerentes com a complexidade do seu programa (ou cargo). Essa complexidade determina o peso relativo da atividade e a sua remuneração por consequência.

Quanto mais habilitados e mais competentes os indivíduos, maior sua empregabilidade, demandando remuneração compatível para a sua maturidade.

No quadro abaixo é possível verificar a relação entre o Plano de Cargos e Salários e a Gestão por Competência.

Tabela 14.4.2 Relação de Planos de Cargos e Salários Tradicional X Gestão por Competência.

Plano de Cargos Tradicional	Gestão por Competência
Voltado ao que deveria ser feito	Foco no que é entregue "O quanto realiza e o quanto agrega valor"
Cargos alocados por funções e por área	Profissionais alocados em eixos de carreiras

Carreiras desenhadas de acordo com a estrutura organizacional	Carreira desenvolvida de forma não atrelada à estrutura organizacional
Difícil mudança de área	A mudança pode ocorrer sem alteração do eixo da carreira
Descrições de funções que se alteram constantemente	Descrições de nível de complexidade (mais estáveis e abrangentes)
Dificulta a flexibilidade funcional, pois as pessoas se prendem ao que deveriam fazer	Total flexibilidade, incentivando a multifuncionalidade
Independente dos objetivos organizacionais	As competências derivam da estratégia organizacional
Número excessivo de cargos/funções	A tendência é de redução e racionalização das funções
Não relaciona perfeitamente desenvolvimento aos cargos	O desenvolvimento, ou capacidade de entrega, é a base para o posicionamento na carreira

De acordo com os autores, as etapas para o bom desenvolvimento do modelo de gestão por competência são as seguintes:

- Definição dos perfis de competências para cada carreira e cargo;
- Avaliação do domínio das competências e estabelecimento de objetivos;
- Identificação das competências necessárias ao negócio;
- Estabelecimento de planos individuais para desenvolver pessoas com foco no perfil de competências;
- Remuneração estratégica de acordo com o crescimento individual;
- Estrutura flexível voltada para o desenvolvimento de competências.

Segundo Souza ividem as competências em gerais e específicas, sendo que as gerais são comuns a toda a organização e as específicas voltadas à carreira. [Souza 2007]

14.4.3 Avaliação de Desempenho

Uma das formas conhecidas para a avaliação de desempenho é a APPO – Avaliação Participativa por Objetivos. Esta consiste de seis etapas que serão citadas a seguir:

- Comprometimento pessoal
- Quanto ao alcance dos objetivos conjuntamente formulados
- Negociação com o Gerente
- Alocação dos recursos e meios necessários para o alcance dos objetivos (recursos técnicos e físicos)
- Desempenho
- Comportamento do avaliado para efetivar o combinado
- Constante monitoração dos Resultados e Comparação com os objetivos formulados
- Verificação dos CUSTOS X BENEFÍCIOS
- Retroação intensiva e contínua
- Avaliação conjunta

14.5 Liderança

O processo de liderança é bastante complexo. Por isso, seus estudos costumam basear-se em diferentes abordagens, sendo que três delas são as mais discutidas. A primeira vê a liderança como uma combinação de traços pessoais. A segunda enfatiza o comportamento do líder. E a terceira pressupõe que as condições que determinam a eficácia da liderança variam de acordo com a situação.

14.5.1 O papel do líder

Mais do que os administradores de pessoal e recursos humanos, os gestores de pessoas têm que desempenhar o papel de líder. Num contexto de Gestão de Pessoas, é necessário que elas vejam-se mais como colaboradores do que como subordinadas, pois se requer sua adesão aos objetivos, políticas e missão da organização. Isso significa que os gestores precisam permanentemente atuar como líderes, já que liderança nada mais é do que a forma de direção baseada no prestígio pessoal e na aceitação dos subordinados.

Segundo o PMBOK, “liderar envolve estabelecer uma direção, desenvolvendo uma visão do futuro e estratégias para se atingir esta visão; alinhar pessoas, comunicando em palavras esta visão; motivar e inspirar as pessoas buscando superar as barreiras” e “gerenciar consiste em sistematicamente produzir resultados esperados para o projeto”. Cabe ao Gerente de Projeto atuar como líder do projeto, mas, esta liderança deve ser demonstrada também por outros membros da equipe em outros níveis, como liderança técnica, por exemplo.

Peter Drucker dizia que todos os líderes eficazes que ele encontrou sabiam quatro coisas simples:

- Líder é alguém que possui seguidores.
- Algumas pessoas são pensadoras, outras profetas, outras realizadoras, etc.
- Todos os papéis são importantes e muito necessários. Mas, sem seguidores, não podem existir líderes.
- Um líder eficaz não é alguém amado e admirado. É alguém cujos seguidores fazem as coisas certas. Popularidade não é liderança; resultados sim.
- Os líderes são bastante visíveis. Portanto, servem de exemplo.

Comment [LA378]: Paragrafo a partir do Segundo deve haaver um recuo de 1,27 cm.

No ambiente de negócios, consultoria e gerenciamento de projetos, atuar com líder implica em desempenhar os seguintes papéis:

- Facilitar a comunicação entre clientes, fornecedores e time de projeto, obtendo *inputs* e comprometimento.
- Buscar soluções inovadoras para os problemas dos clientes.
- Aumentar a autonomia e participação dos membros do projeto nos processos de planejamento, decisão, resolução de problemas e gerenciamento.
- E estabelecer padrões de excelência de performance e produtividade através do uso da tecnologia.

Segundo Magnos, uma pesquisa da empresa de consultoria Cambria Consulting estudou as competências de liderança de grandes empresas mundiais como AT&T, British Petroleum, DuPont, Ford, GE, Hewlett-Packard, Johnson & Johnson, PepsiCo, Siemens e Unilever, com o intuito de procurar identificar as práticas e atributos pessoais mais desejáveis em um líder.

Esta pesquisa resultou em :

Tabela 14.5.1 Práticas e atributos pessoais desejáveis em um líder.

Práticas Desejáveis	Percentual
Integridade/Sinceridade/Ética	77%
Desenvolver pessoas	64%
Capacidade de Realização	76%
Obter resultados	55%
Habilidade de lidar com pessoas	73%
Concentrar-se no cliente	52%
Orientação para o aprendizado	73%
Comunicar-se	52%

14.5.2 Coach

O termo *coach* teve origem no campo esportivo e tem sido utilizado para designar

muito utilizado nas empresas, não necessariamente para designar um cargo, mas um papel: o da pessoa que se compromete a apoiar alguém com vista em melhorar seu desempenho e promover seu desenvolvimento profissional e pessoal.

Segundo Araujo é um profissional que não se compromete apenas com os resultados, mas com o próprio desenvolvimento e realização da pessoa que é designada como seu cliente. Sua função é dar-lhe poder para que ela produza, para que suas intenções se transformem em ações que, por sua vez, se traduzam em resultados, Araujo 1999. 'a ação desse profissional é denominada *coaching*, ou seja, processo de desenvolvimento com as pessoas e que tem por finalidade principalmente encorajar o crescimento pessoal e profissional fornecendo suporte e estrutura para se alcançar objetivos e obter resultados.

Assim, o funcionário passa a ter maior produtividade, também reduz os conflitos dentro dos grupos ou equipes, pois, com o *coach* tende a ficar mais claro para cada profissional, qual o papel dele na equipe.

14.6 Gerenciamento de Conflitos

Conflitos é um grande desafio na área de gestão de pessoas, estes são praticamente inevitáveis, pois, toda vez que algo for mudado existem grandes chances de gerar muitos conflitos. E o grande desafio é como aplicar à gerência de conflitos de tal forma que seja positiva e imprescindível a atuação do grupo fazendo com que este se torne viável, autocrítico e criativo no ambiente de trabalho.

14.6.1 Visão dos Conflitos

Segundo Verman 1996 existem três diferentes pontos de vista acerca dos conflitos que podem ocorrer entre pessoas ou entre grupos, são elas:

- **Visão Tradicional:** esta enfatiza que todo conflito é ruim, portanto, deve ser evitado, visto este como uma disfunção resultante de falhas de comunicação, abertura e confiança. Era consistente com as atitudes de grupo que prevaleciam nas décadas de 30 e 40.
- **Visão Contemporânea:** o conflito é uma consequência natural e inevitável em qualquer grupo. Não sendo necessariamente ruim, podendo ter o potencial de ser uma força positiva. Perdurou na década de 40 até a metade da década de 70.
- **Visão Integracionista:** nesta o conflito pode ser uma força positiva. Defende abertamente a tese de que algum conflito é absolutamente necessário para o desempenho eficaz de um grupo.

14.6.2 Níveis de Conflito

Existem quatro níveis de conflito:

- **Conflitos internos:** ocorre quando duas ou mais opiniões opostas ocorrem em um único indivíduo.
- **Conflitos entre indivíduos:** os conflitos entre indivíduos dentro da organização são vistos como resultados de diferenças de personalidade.
- **Conflitos entre indivíduos e grupos:** o indivíduo que não concorda com

cultura organizacional estará em conflito com o grupo de trabalho ou com toda a organização.

- **Conflitos entre grupos:** o conflito entre grupos é inevitável devido à competição por recursos escassos e pelos diferentes estilos gerenciais necessários para a operação eficaz de diferentes departamentos.
- **Conflitos entre organizações:** onde cada empresa procura o dinheiro do consumidor no mercado, e essa competição leva as organizações a entrarem em conflito.

14.6.3 Conflito e Estresse

O estresse está diretamente ligado a conflitos, pois, todo conflito sempre gera certo grau de estresse. E estresse pode fazer com que pessoas tenham dores de cabeça, úlceras, pressão alta, acidente vascular cerebral e infarto. Por isso, esse é um dos fatores primordiais que devem ser observados e evitados no ambiente de trabalho, pois funcionário estressado é sinônimo de baixa qualidade.

E nem sempre o estresse é causado por excesso de demandas, muitas vezes também ocorre, pois as demandas impostas ao indivíduo são muito menores do que suas potencialidades, podendo assim ter um efeito negativo, da mesma forma.

Sabe-se que estresse afeta as relações sociais e o desempenho profissional, empobrece nossa capacidade de ouvir o outro, enfraquece nossa concentração, fatores estes que favorecem o retraimento do espírito de equipe, típicos do esgotamento total. Porém, Gmelch 1993, baseando-se no gap de que estresse é um fato normal da vida, afirma que o mais importante é aprender a usá-lo construtivamente para melhorar o desempenho.

Mas, para se chegar a esse nível sabe-se que é preciso um alto nível de gerenciamento de estresse.

14.6.4 Como Gerir Conflitos no Ambiente do Projeto

Um estudo realizado pela American Management Association em 1996, concluiu-se que o gerente de projetos, gasta em média 20% de seu tempo tratando do gerenciamento de conflitos.

Um efetivo gerenciamento de conflitos, requer em primeiro lugar, verificar habilidades para entender e diagnosticar corretamente sua causa; em seguida, de acordo com as personalidades das pessoas envolvidas no conflito, verificarem quais estratégias de comunicação e negociação utilizar, para que se tenha um clima de confiança e respeito.

A tabela abaixo apresenta algumas técnicas para a resolução de conflitos e seus possíveis efeitos, estas foram apresentadas por Blake e Mounton [citados por VERMA 1996].

Tabela:

Estilo	Descrição	Efeito
Retirar-se temporariamente	Retirar-se de uma relação ou potencial situação de conflito temporariamente.	Não resolve o problema
Mudar o foco	Enfatizar as áreas de concordância em detrimento	Fornecer apenas uma resolução de curto prazo

Empregar a força	Forçar seu ponto de vista independentemente dos outros; oferece soluções do tipo ganha-perde	Sentimentos de vingança podem voltar de outra maneira
Firmar um compromisso	Procurar uma troca que traga algum grau de satisfação para todas as partes	Fornece uma resolução definitiva do conflito
Resolver o problema	Trata conflito como um problema a serem resolvidos, examinando-se todas as possíveis alternativas; requer diálogo aberto e atitude proativa dos dois lados.	Fornece uma resolução de longo prazo
Negociar	Incorporar todos vários pontos de vista e visões das diferentes perspectivas; conduz a um consenso e compromisso.	Fornece uma solução de longo prazo

Fonte:????????????????????????????

14.7 Gestão de Pessoas e Desenvolvimento da Inteligência Emocional

Atualmente, as habilidades emocionais vem se destacando como um dos fatores cruciais para o perfil do profissional de sucesso, valorizando no indivíduo, as características, peculiaridades e habilidades próprias, pois os gestores de pessoas habilidosos sabem que os profissionais bem-sucedidos são aqueles que encontram o equilíbrio entre razão e emoção. [Albuquerque 2009]

Segundo Goleman, a inteligência emocional pode ser compreendida através de uma descrição resumida dos traços e características que são consideradas como as mais importantes na vida daqueles que são os melhores profissionais ou que exibem um desempenho elevado.

Esses traços são característicos de dois tipos:

- **Os que dizem respeito à competência pessoal:** que são o autoconhecimento e o autocontrole.
- **Os que dizem respeito à competência social:** que são a empatia e as aptidões interpessoais.

Ou seja, ter inteligência emocional e possuir a capacidade de criar motivações para si e de persistir num objetivo apesar dos percalços; de controlar impulsos, de se manter em bom estado de espírito, de impedir que a ansiedade interfira na capacidade de raciocinar; de ser empático e autoconfiante. Existindo muitas evidências que atestam que as pessoas emocionalmente competentes levam vantagem em qualquer setor da vida, seja nas relações amorosas e íntimas, seja assimilando as regras táticas que governam o sucesso da política organizacional. Já aquelas que não conseguem exercer

Sendo assim um dos maiores problemas dos gestores e conseguir encontrar o equilíbrio entre pessoas e organizações. Levando em conta capital intelectual, as características de cada um, onde uns são mais autoconfiantes, outros não; cultura familiar que foram criados, enfim que uns profissionais sabem lidar muito bem com suas próprias emoções e outros não sabem muito bem lidar com as próprias emoções e nem com as emoções dos outros. Cabe a cada gestor começar com um levantamento minucioso sobre as características de cada pessoa de sua equipe, para assim poder guiar melhor suas atitudes e obter maior satisfação e equilíbrio na relação empresa/trabalhadores e assim obter um maior índice de sucesso. Facilitando também o desenvolvimento tanto das competências corporativas, como as humanas.

14.7.1 Conhecimentos, Habilidades e Atitudes (CHA)

A procura por profissionais que agreguem valor a empresa através do seu conjunto de competências e que estes consigam identificar-se com os seus objetivos de sucesso mutuo, 'e o desejo de todo gestor. E para que um profissional possua estas características 'e preciso que o mesmo tenha conhecimento, seja hábil e queira realizar o trabalho.

Assim CHA 'e o conjunto de conhecimentos, habilidades e atitudes que credenciam um profissional a exercer determinada função.[ABRH-RN]

A tabela abaixo apresenta o principal foco de cada característica.

Tabela 14.7.1 – Significado de C.H.A

C	H	A
Conhecimento	Habilidade	Atitude
Saber	Saber Fazer	Querer Fazer

Fonte:

E de forma mais detalhada abaixo será discorrido sobre o significado de cada uma:

- **Conhecimento:** saber aprimorar, tornar claro aquilo que ainda não se conhece, ou que deseja conhecer mais afundo. O conhecimento é a base de tudo. O conhecimento é adquirido de várias formas, nas ruas, nas faculdades, na internet, nos livros, com a vida e etc;
- **Habilidade:** Praticar o que conhece, saber fazer. Todo conhecimento que temos é aperfeiçoado com a Habilidade;
- **Atitude:** Querer fazer, arriscar, se comprometer. É certo que o conhecimento e a habilidade são fatores muito importantes, mas e a atitude? Também é definida como sendo uma predisposição ou tendência de responder, positivamente ou negativamente, a determinados fatos, idéias, objetos, pessoas ou situações.

Ou seja, de nada adiantam conhecimentos e habilidades, caso não sejam

E para isso 'e preciso desenvolver a capacidade de Transformar Conhecimentos, Habilidades e Atitudes em resultados, onde, esta pode ser designada competência. Existem dois tipos de competência:

- **Competência Cooperativa:** é um conjunto de qualificações e tecnologias da instituição, necessárias ao alcance dos objetivos estratégicos.
- **Competência Humana:** conjunto de conhecimentos, habilidades e atitudes que geram resultados para a organização e agregam valor à pessoa.

Onde o C e o H caracterizam as competências técnicas (tudo que o colaborador precisa ter para desempenhar bem o seu papel) e o A as competências comportamentais (o diferencial competitivo de cada profissional que tem impacto nos seus resultados). As competências comportamentais são identificadas a partir das competências institucionais. Por isso, estão relacionadas diretamente aos objetivos estratégicos da instituição. Portanto, 'e de suma importância levá-las em consideração no momento de formação da equipe, conseqüentemente devem ser utilizadas nos processos de seleção.

Assim, 'e necessário criar um plano, para manter um banco de dados, que armazene as competências de cada pessoa que passar pela empresa, para que se possa ter uma equipe mais eficiente.

14.8 Considerações Finais

<FALTA COMPLEMENTAR>

14.9 Sugestões de Leitura

BRANDEN, Natahaniel. *Auto-estima no trabalho: como as pessoas confiantes e motivadas constroem organizações de alto desempenho*. Rio de Janeiro: Campus, 1999.

Trata-se de um guia que os gerentes podem utilizar para favorecer a auto-estima e a motivação dos empregados. Em seu último capítulo apresenta um programa de desenvolvimento auto dirigido.

COVEY, Stephen R. *Liderança baseada em princípios*. Rio de Janeiro: Campus, 2002.

FARIA, Carlos. Alberto. *Alinhamento Pessoal*
http://www.merkatus.com.br/11_artigos/AlinhamentoPessoal.htm . Acesso em Novembro, 2009.

Este artigo 'é muito interessante pois, faz com que as pessoas se questionem sobre as posturas que adotam e tem adotado ao longo da vida, fazendo com que reflitam se 'é isso que querem realmente ser. Podendo assim ser muito útil para que muitas pessoas possam melhorar sua conduta e postura de vida, podendo evoluir como profissional.

FARIA, Carlos Alberto. *Lei da Expectativa Negociada*
http://www.merkatus.com.br/20_leis/18.htm . Acesso em Novembro, 2009.

Outro artigo muito interessante, pois, muitos de conflitos gerados em ambientes de trabalho, e falta de motivação dos funcionários, existe por que ambos empresa/trabalhador, não correspondem em expectativas. E na maioria das vezes ambos, não expuseram as expectativas esperadas, logo fica praticamente impossível o outro realiz'á'-la.

MAXWELL, John.C. *O livro de ouro da liderança*. Rio de Janeiro: Thomas Nelson Brasil, 2008.

Destaca as principais ações de um líder, e mostra como os ensinamentos de uma boa liderança podem ser aprendidos por qualquer pessoa. Discorre sobre varias lições fundamentadas em uma vida de experiências como líder de destaque.

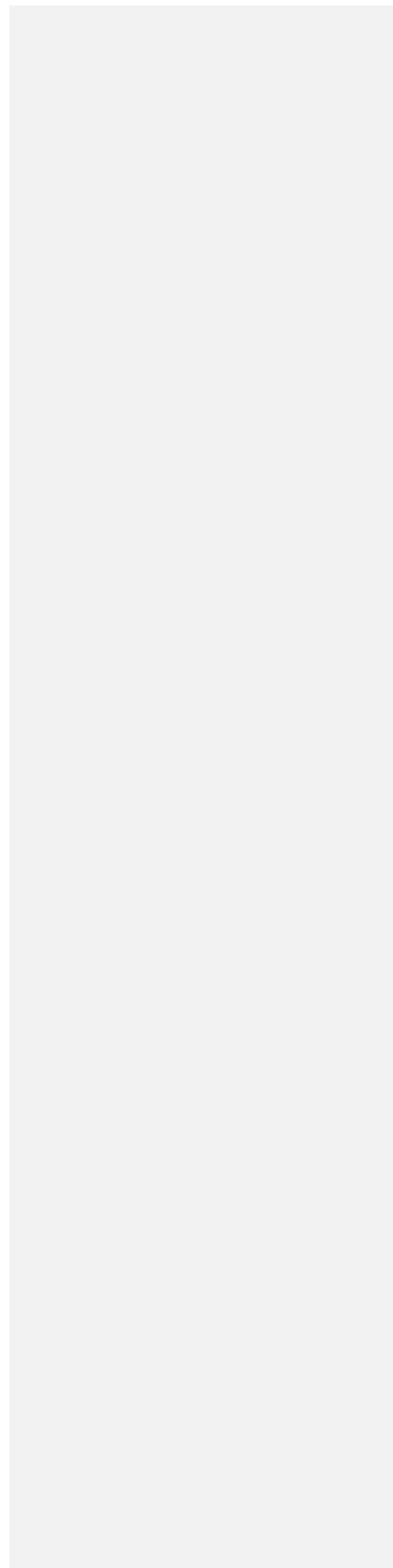
ULRICH⁶⁷, Dave. *Os campeões de recursos humanos: inovando para obter os melhores resultados*. São Paulo: Futura, 1998.

Este livro analisa em seu primeiro capítulo oito desafios competitivos enfrentados atualmente pelas empresas e demonstra como elas precisam gerar novas capacidades organizacionais que derivem de redefinição e redistribuição das praticas e funções profissionais de RH para enfrentar estes desafios.

14.10 Tópicos de Pesquisa

<FALTA COMPLEMENTAR>

Pessoas e metodologias ágeis
Pessoas e Inovação



14.11 Exercícios

1. Discuta a afirmação: “O ser humano é movido por necessidades.”
2. Identifique situações no trabalho que podem satisfazer às necessidades de estima.
3. Justifique a necessidade de conhecimento da cultura organizacional para que se possam promover mudanças significativas nos processos administrativos de uma empresa.
4. Analise o comportamento de um líder em relação aos seus subordinados e defina se corresponde a Teoria X ou a Teoria Y.
5. Em que medida a ação do *coach* pode ser benéfica para ao trabalho em equipe?
6. O que uma empresa pode fazer para que seus empregados vistam a camisa da empresa?
7. Justifique a necessidade de Gerenciamento de Conflitos e discuta sobre até que ponto se deve intervir em um conflito.
8. Justifique o porquê a técnica *downsizing* é tão criticada pelos gurus da Gestão de Pessoas.
9. Quais os fatores que se destacam no Gerenciamento de Pessoas?
10. Qual a postura que deve ser adotada por um líder eficiente?
11. Quais as vantagens da Gestão por Competência?

Referências

- ARAÚJO. A, "Coach: um parceiro para o seu sucesso."
- BOYETT, J. H.; BOYETT, J.T. O guia dos gurus: os melhores conceitos e práticas de negócios. Rio de Janeiro: Campus, 1999.
- DUTRA. S.J, "ADMINISTRAÇÃO DE CARREIRAS: Uma Proposta para Repensar a Gestão de Pessoas" – 176pgs - 1ª Ed – 1996
- CAPELLI. P, "Contratando e Mantendo as Melhores Pessoas". Rio de Janeiro: Record, 2003.
- CASTELIANO, Tania, Você sabe ouvir. Rio de Janeiro: Best Seller, 2009.
- CHIAVENATO, I. Introdução à teoria geral da administração.
- CHIAVENATO, I. "Gerenciando Pessoas". -3.ed.-São Paulo: Makron Books,1994.
- DAVEL, Eduardo. VERGARA Sylvia Constant. (Org.) "Gestão com pessoas e subjetividade". São Paulo: Atlas, 2001.
- FLEURY, M. T. L. FISCHER, R. M. Cultura e Poder nas Organizações . 1 ed. São Paulo : Atlas, 1992.
- GIL. A.C, "Gestão de Pessoas". São Paulo: Atlas, 2009
- GOLEMAN. D, "Inteligência Emocional".
- GRUN. A ASSLANDER. F, "A Arte de Ser Mestre de Si Mesmo Para Ser Líder de Pessoas". Petropolis, RJ: Vozes, 2008.
- KING. W.J, "As Regras de Ouro da Conduta Profissional." Rio de Janeiro: Sextante, 2008.
- KOHN. S.E, CONNELL. V.D. " O Gerente Eficiente". São Paulo: Landscape, 2008.
- KOONTZE1980
- MAGNUS. S, "Liderança Práticas e Atributos."
- MANÃS, A. V, "Gestão por competências".
- MAX. G, "Gestão de Pessoas".
- MAXWEL. J, O Livro de Ouro da Liderança. Rio de Janeiro: Thomas Nelson Brasil, 2008.
- MAYO. A, "O Valor Humano da Empresa". São Paulo: Prentice Hall, 2003.
- MCGREGOR. D. "O Lado Humano da Empresa". Terceira Edição. São Paulo: Martins Fontes, 1999.
- NERY.A.M et al. "Pensamento Criativo: Uma Ferramenta Para Gerenciamento De Projetos." Fundação Getúlio Vargas – 2007
- OLIVEIRA, Otávio J. (Org.). Gestão empresarial: sistemas e ferramentas. São Paulo: Atlas, 2007.
- PMI 2004
- RAJ.P.P et al. "Gerenciamento de Pessoas em Projetos". Rio de Janeiro. Editora FGV Management, 2008.
- Revista Melhor 2008
- SANTOS. M, "Práticas de Gerenciamento Estratégico da Informação: Como as Empresas Brasileiras Estão Utilizando a Informação para a Competitividade." Universidade Anhembi-Morumbi SP -Fundação Instituto de Administração – USP
- SANTOS. R. A, "A criação de vantagens competitivas através da gestão estratégica a informação."
- SOTILLE. M.A, "Ética e Responsabilidade Profissional em Gerência de Projetos."

SILVA. A.H. N, "Gestão do Conhecimento: Conceitos, Ferramentas e Aplicações." Centro de Informática – Universidade Federal de Pernambuco (UFPE)
TRAVASSOS. Os "Recursos Humanos: como gerenciá-los em projetos?". Mundo PM
VERMAN
WOOD. T. Jr, FILHO. V.P. "Remuneração e Carreira: por Habilidades e Competências.".-
3.ed.- São Paulo: Atlas,2004.

Índice do Capítulo

14.1. PROCESSO DE COMUNICAÇÃO ERROR! BOOKMARK NOT DEFINED.

14.1.1. A COMUNICAÇÃO ERROR! BOOKMARK NOT DEFINED.

14.1.2. MODELO DE COMUNICAÇÃO ERROR! BOOKMARK NOT DEFINED.

14.1.3. CANAIS DE COMUNICAÇÃO ERROR! BOOKMARK NOT DEFINED.

14.1.4. A COMUNICAÇÃO EM ORGANIZAÇÕES ERROR! BOOKMARK NOT DEFINED.

14.1.5. COMUNICAÇÃO EM PROJETOS ERROR! BOOKMARK NOT DEFINED.

14.1.6. A COMUNICAÇÃO COMO DESAFIO PARA O GERENTE DE PROJETO
ERROR! BOOKMARK NOT DEFINED.

14.2. GERENCIAMENTO DE COMUNICAÇÃO EM PROJETOS ERROR! BOOKMARK NOT DEFINED.

14.2.1. PLANEJAMENTO DAS COMUNICAÇÕES ERROR! BOOKMARK NOT DEFINED.

14.2.2. DISTRIBUIÇÃO DAS INFORMAÇÕES ERROR! BOOKMARK NOT DEFINED.

14.2.3. RELATÓRIO DE DESEMPENHO ERROR! BOOKMARK NOT DEFINED.

14.2.4. GERENCIAR AS PARTES INTERESSADAS ERROR! BOOKMARK NOT DEFINED.

14.3. CONCLUSÕES ERROR! BOOKMARK NOT DEFINED.

14.4. TÓPICOS DE PESQUISA ERROR! BOOKMARK NOT DEFINED.

14.5. SUGESTÕES DE LEITURA ERROR! BOOKMARK NOT DEFINED.

14.6. EXERCÍCIOS ERROR! BOOKMARK NOT DEFINED.

REFERÊNCIAS ERROR! BOOKMARK NOT DEFINED.

Capítulo

14

Gestão da Comunicação

O capítulo visa por meio de uma referência didática contribuir para a ampliação do conhecimento e auxiliar pessoas que necessitem aplicar, de forma eficaz, o processo de comunicação em projetos de software. Este capítulo aborda uma visão geral da comunicação, dos processos da Gerência de Comunicação de Projetos, bem como sugestões de leitura, tópicos de pesquisa e exercícios. Inicialmente serão abordadas questões referentes ao processo da comunicação em geral, em torno da sua definição, importância, seus elementos básicos e aspectos do uso da comunicação em organizações e projetos, como a comunicação representa um desafio para o gerente, concluindo com o gerenciamento da comunicação em projetos sendo detalhados seus respectivos processos.

Visão Geral da Comunicação

Nossas sociedades primitivas desenvolveram a fala por necessidade de comunicação e troca de informações. No entanto, não havia uma forma de registrar o que era falado, e então desenvolveram símbolos e desenhos para deixarem o registro das suas informações, as quais são encontradas atualmente nos sítios arqueológicos, assim, originando às primeiras formas escritas de comunicação. Segundo Chaves [Chaves et. al. 2006], a partir da prensa de Gutenberg a comunicação escrita se multiplicou para todos. Posteriormente surgiram outros meios de divulgação como o código Morse, o telefone de Bell e o rádio de Marconi, o que possibilitou a comunicação universal entre as pessoas.

Atualmente, dentre todas as formas nas quais a comunicação é utilizada vale destacar como esta é empregada nas empresas e/ou organizações. Segundo Maron [Maron 2008], uma organização nada mais é do que a reunião de pessoas integradas e constantemente se comunicando, a serviço de outras pessoas. Quando esta comunicação existe e é feita com qualidade e profundidade, abrem-se portas para soluções de problemas e dificuldades com simplicidade e criatividade, permitindo decisões com segurança e rapidez, atingindo os melhores resultados. Um ambiente aberto à comunicação permite que as pessoas se sintam respeitadas e satisfeitas por contribuírem e participarem ativamente. O resultado será sempre a conquista de maior produtividade, progresso para todos e resultados positivos em todos os níveis. A boa comunicação é primordial para o sucesso de quaisquer projetos e consequentemente da organização, pois além de todos os seus benefícios também “contamina” as pessoas com a alegria e o otimismo.

Os projetos das organizações são realizados por pessoas, as quais necessitam incondicionalmente da comunicação para cumprir os objetivos estabelecidos e consequentemente compreender como as tarefas devem ser realizadas nos projetos. Assim, a comunicação é um elemento essencial no gerenciamento de qualquer projeto, pois utiliza recursos de troca e partilha capazes de promover a compreensão mútua.

Comment [WU379]: Antes da descrição do capítulo faltou o nome da autora!

Comment [WU380]: Acho que esse parágrafo ficou muito grande, poderia ser melhor dividido pra facilitar o entendimento.

Comment [WU381]: Acho que devia ser numerada como seção 14.1

Comment [WU382]: O trema foi extinto de acordo com a nova ortografia da língua portuguesa

Comment [WU383]: Idem ao comentário 4

14.1. Processo da Comunicação

14.1.1. A Comunicação

O conceito de comunicação surgiu com Aristóteles, ao dizer que o processo necessita de três elementos: aquele que fala, o que essa pessoa tentou dizer e aquele que escuta, criando assim o modelo de comunicação utilizado até os dias de hoje.

Conforme Mendes [Mendes 1999], a comunicação significa tornar comum, trocar informações, partilhar idéias, sentimentos, experiências, crenças e valores por meio de gestos, palavras, atos, imagens, símbolos, figuras, entre outros. Comunicação palavra de origem latina (*communicare*) cujo significado é tornar comum, partilhar, repartir, associar, trocar opiniões, conferenciar. A comunicação implica em participação, em troca de mensagens, em emissão ou recebimento de informações. A comunicação é a provocação de significados comuns, com suas reações resultantes, entre comunicador e intérprete, por meio do uso de sinais e símbolos.

A comunicação constitui uma das ferramentas mais importantes que os líderes têm à sua disposição para desempenhar as suas funções de influência. A sua importância é tal que alguns autores a consideram mesmo como o “sangue” que dá vida à organização. Esta importância deve-se essencialmente ao fato de apenas por meio de uma comunicação efetiva ser possível:

- Estabelecer com a participação de membros de todos os níveis hierárquicos da organização, os objetivos organizacionais de forma que contemplem, não apenas os interesses da mesma, mas também de todos os seus membros.
- Definir, com a participação de membros de todos os níveis hierárquicos da organização, a estrutura organizacional, sendo ao nível do desenho organizacional, da distribuição de autoridade, responsabilidade e tarefas.
- Definir com a participação dos membros da organização, desde o nível do desenho organizacional até a distribuição de autoridade, responsabilidade e tarefas. Coordenar, fornecer apoio e controlar as atividades de todos os membros da organização.
- Efetuar a integração dos diferentes departamentos e permitir a ajuda e cooperação interdepartamental.
- Desempenhar eficazmente o papel da influência por meio da compreensão e atuação em conformidade com a satisfação das necessidades e sentimentos das pessoas com a finalidade de aumentar a sua motivação.
- Além da importância que a comunicação assume nas organizações é fundamental destacar também, as suas principais funções conforme mostra a Tabela 14.1.

Comment [WU384]: Acho que esse “com” deve ser excluído, ou então ser colocada uma vírgula antes do “com”

Comment [WU385]: Acho que a concordância correta é “responsabilidades”

Comment [WU386]: Idem ao comentário 6

Comment [WU387]: Texto praticamente idêntico ao do tópico anterior, acho que o texto deveria ser excluído ou alterado.

Tabela 14.1. Funções da comunicação na organização.

FUNÇÃO	DESCRIÇÃO
Controle	Através das hierarquias de autoridade e orientações formais.

Comment [WU388]: Colocar a legenda na mesma página que a tabela.

	desempenho e como melhorá-lo
Expressão Emocional	Integração social dentro de grupos satisfazendo as necessidades sociais.
Informação	Fornecer subsídios para facilitar a tomada de decisão

14.1.2. Modelo de Comunicação

Para o desenvolvimento de políticas de comunicação eficazes é necessário analisar antes cada um dos elementos que fazem parte do processo de comunicação. Assim, fazem parte do modelo do processo de comunicação o emissor, um canal de transmissão, geralmente influenciado por ruídos, um receptor e ainda o *feedback* do receptor conforme mostra a Figura 14.1 [Cavaliere 2005].

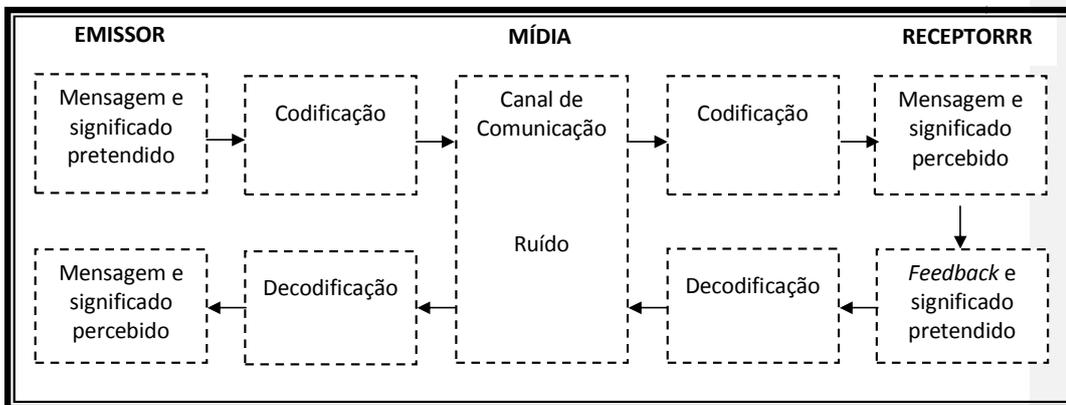


Figura 14.1. Processo de Comunicação [Adaptada de Cavaliere 2005].

Comment [WU389]: Figura está ultrapassando as margens determinadas do texto

O emissor (ou fonte da mensagem da comunicação) é o componente que representa quem pensa, codifica e envia a mensagem, ou seja, quem inicia o processo de comunicação. A codificação da mensagem pode ser feita transformando o pensamento que se pretende transmitir em palavras, gestos ou símbolos que sejam compreensíveis por quem recebe a mensagem.

A mensagem significa conteúdo, aquilo que é dito, escrito ou transmitido por símbolos ou sinais, e seu objetivo é gerar reações e comportamentos. Pode ser transmitida pela voz, por um texto, por meio de um desenho, por gestos, movimentos e expressões faciais ou por meios eletrônicos.

O canal de comunicação faz a ligação entre o emissor e o receptor e representa o meio através do qual é transmitida a mensagem. Existe uma grande variedade de canais de transmissão, cada um deles com vantagens e desvantagens: destacam-se o ar (no caso do emissor e receptor estarem frente a frente), o telefone, os meios eletrônicos, os memorandos, a rádio, a televisão, entre outros.

Receptor da mensagem representa quem recebe e decodifica a mensagem.

Portanto, neste momento é necessário ter atenção no processo de decodificação da mensagem, a qual resulta efetivamente no que emissor pretendia enviar (por exemplo, em diferentes culturas, um mesmo gesto pode ter significados diferentes). Podem existir apenas um ou numerosos receptores para a mesma mensagem.

Os ruídos são obstruções mais ou menos intensas ao processo de comunicação e

Comment [WU390]: Acho que faltou um "o" e o certo é "que o emissor"

no canal de transmissão. Obviamente estes ruídos variam de acordo com o tipo de canal de transmissão utilizado, as características do emissor e do receptor, sendo, por isso, um dos critérios utilizados na escolha do canal de transmissão.

O *feedback* ou realimentação é a resposta do receptor ao emissor da mensagem e pode ser utilizada como uma medida do resultado da comunicação, para se certificar de que a interação está sendo mantida no momento em que a mesma está se processando, e ajuda no processo de conhecimento para saber se a mensagem foi enviada, como foi recebida e se foi ou não compreendida. Pode ou não ser transmitida pelo mesmo canal de transmissão.

Comment [WU391]: Parágrafo ficou um pouco extenso, tentar reduzir para facilitar o entendimento.

14.1.3. Canais de Comunicação

A comunicação é utilizada em vários canais, cada um deles apresentando vantagens e desvantagens em seu uso. Entre os canais utilizados podem ser citados os orais, os escritos, os eletrônicos e os digitais.

A comunicação oral possui como principal característica a presença do receptor (exclui-se, obviamente, a comunicação oral que utilize a televisão, a rádio, ou as gravações). Esta característica explica diversas das suas principais vantagens, nomeadamente o fato de permitir o *feedback* imediato, a passagem imediata do receptor ao emissor e vice-versa. A utilização de comunicação não verbal como os gestos, a mímica e a entonação, por exemplo, facilita as retificações e explicações adicionais, observar as reações do receptor, e ainda a grande rapidez de transmissão. Como principais desvantagens da comunicação oral destacam-se o fato de ser efêmera, não permitindo qualquer registro e, conseqüentemente, não se adequando a mensagens longas e que exijam análise cuidadosa por parte do receptor.

Comment [WU392]: Idem ao comentário 4.

A comunicação escrita teve o seu auge, e ainda hoje predomina, nas organizações burocráticas que seguem os princípios da Teoria da Burocracia enunciados por Max Weber. A sua principal característica é o fato do receptor estar ausente tornando-a, por isso, um monólogo permanente do emissor. Como principais vantagens da comunicação escrita, podemos destacar o fato de ser duradoura, permitir um registro, além de exigir uma maior atenção à organização da mensagem, sendo assim adequada para transmitir políticas, procedimentos, normas e regras. Adequa-se também a mensagens longas e que requeiram uma maior atenção e tempo por parte do receptor tais como relatórios e análises diversas.

Como principais desvantagens destacam-se a referida ausência do receptor, o que impossibilita o *feedback* imediato, não permite correções ou explicações adicionais e obriga ao uso exclusivo da linguagem verbal.

A comunicação por meios eletrônicos e digitais permite a transmissão de um grande volume de informação de maneira veloz e precisa. A internet, o correio eletrônico e a telefonia celular são exemplos dessa forma de comunicação, em que é necessária a tecnologia dos computadores e das telecomunicações para se concretizar. Permite desde o simples envio de uma mensagem de texto à transmissão de uma videoconferência, com imagens dos emissores e receptores interagindo, conversando e debatendo ao vivo. No entanto, necessita da existência e correto funcionamento de uma infra-estrutura tecnológica de suporte.

14.1.4. A Comunicação em Organizações

Atualmente, o ambiente organizacional é caracterizado por mudanças contínuas, assim, surgindo a necessidade de mudança nos modelos tradicionais das práticas da comunicação organizacional para manter a competitividade empresarial.

A comunicação em organizações é utilizada como estratégia competitiva de mercado, englobando todas as formas de comunicação utilizadas para alcançar e interagir com seus

ser **trabalha** de forma convergente com os propósitos que pretende alcançar em curto, médio e longo prazo.

Comment [WU393]: Acho que a palavra correta seria "trabalhada" e não "trabalha"

Ao trabalhar a comunicação em organizações a empresa fortalecerá a marca corporativa e sua reputação, valores estes que atualmente são relevantes e influenciam na decisão de compra do produto ou serviços oferecidos. Para que a comunicação atinja seus objetivos na organização é necessário que seja transparente e que tenha credibilidade, o que só pode ser alcançado se o discurso for condizente com a prática diária e que seja percebido por seus públicos internos e externos.

Segundo Ribeiro [Ribeiro 2004], uma empresa depende de resultados imediatos, mas não pode alcançá-los sem atentar simultaneamente para sua sustentabilidade e do negócio como um todo. Quando se trabalha visando fortalecimento de marca e reforço de reputação o que está sendo buscado é mais vinculação, compromisso e relações mais profundas entre a organização e seus públicos. Deve-se utilizar a comunicação organizacional de forma integrada o que, em síntese, constitui a somatória dos serviços de comunicação feitos sinergicamente tendo em vista, sobretudo, os públicos a serem atingidos para chegar aos objetivos propostos.

A real eficácia da comunicação é o objetivo fim de um trabalho integrado, o que somente é possível mediante uma ação conjugada de todas as áreas que produzem, emitem e veiculam mensagens aos diferentes públicos de interesse.

Portanto, cada vez mais dentro das empresas, a comunicação deve ser vista como uma estratégia de crescimento. A intensidade do fluxo de informações que existe atualmente exige agilidade e eficiência na comunicação, que se tornou um dos fatores primordiais para o sucesso do projeto, que conseqüentemente, favorece e retorna benefícios para a organização.

14.1.5. Comunicação em projetos

O ato de comunicar em um projeto é muito mais do que simplesmente o envio de um email esporádico ou uma reunião sem pauta e sem ata. Comunicar é a arte que o ser humano tem para se fazer compreender, vender uma idéia, dar uma explicação, ensinar aos outros, transmitir conhecimento, entre outros. A comunicação apóia a tomada de decisões e conquista aliados, mas ao mesmo tempo que, se é mal entendida e mal elaborada, conquista "inimigos".

Para Cavalieri [Cavalieri 2005], os projetos são executados por pessoas. E são elas que resolvem os problemas, tomam as decisões, desenham os modelos, entre outras tarefas. Para serem bem sucedidos, os projetos são obtidos por meio de uma combinação de decisões, ações e estratégias eficazes.

Inicialmente, é necessário saber os objetivos do projeto, quem será o líder ou gerente e os limites do projeto. Basicamente é uma maneira de dizer por que realizar um projeto, a quem se reportar inicialmente e de maneira mais rebuscada dizer 'isso não será feito' ou 'isso não é de responsabilidade deste grupo nesse período'. Outro ponto importante é deixar claro qual o tempo estimado para desenvolver o projeto, qual folga se pode ter em determinada tarefa, quanto se pode gastar nele, quanto do valor pode ser acrescido e em quais circunstâncias. Após a definição e o planejamento de todas as áreas que envolvem o planejamento de um projeto, é o momento de fazer o controle do mesmo, a fim de superar as barreiras da comunicação identificadas.

Comment [WU394]: Acho que é desnecessário esse "o planejamento de"

Barreiras são elementos que interferem e distorcem o processo de comunicação, dificultando ou impedindo o correto entendimento entre o emissor e o receptor. Essas barreiras podem ser de: **conhecimento**, onde inclui o despreparo para lidar com o processo oral ou escrito de comunicação, o uso da linguagem não familiar aos envolvidos e a falta de conhecimento do assunto a ser comunicado; **comportamentais**, quando ocorrem

Comment [WU395]: Acho que o certo é

organizacionais e técnica, quando as estruturas organizacionais são inflexíveis ou excessivamente burocráticas, possuem excesso de regras, procedimentos, padrões e equipamentos inacessíveis ou inadequados.

Comment [WU396]: Acho que o certo é "organizacional"

A comunicação se tornou um insumo estratégico para as empresas, um valor agregado aos seus negócios e **conseqüentemente** uma vantagem competitiva. Em razão destes motivos, as empresas necessitam construir uma identidade corporativa sólida, real, que pode ser realizada por meio da sua comunicação, o que realmente faz e finalmente, como é percebida pelos seus públicos [Maron 2008]. Fazer comunicação exige conhecimento, planejamento, execução e finalmente, mensuração de resultados, principalmente tratando-se de projetos, pois caso contrário não se alcança o objetivo proposto.

Comment [WU397]: Idem 4

- A comunicação eficaz também atua como a "cola" que irá manter unida uma equipe **propiciar o alto desempenho. Mensagens claras são enviadas, recebidas e interpretadas de forma acurada.**

Comment [WU398]: Acho que o certo é "propiciando"

O fato é que, não importa o quanto à tecnologia avance projetos, sempre serão executados por pessoas e, dependerão muito delas para que sejam bem sucedidos. Saber lidar com os desafios da comunicação é um fator crítico de sucesso para o projeto, e uma questão de sobrevivência no mercado para o gerente de projeto. No entanto, ainda existem vários casos em que excelentes profissionais, detentores de sólida formação técnica, por vezes se vêem em dificuldades no exercício da gerência de projetos, porque descobrem que além do perfil técnico, precisam por em prática uma série de habilidades para as quais não estão devidamente preparados, tais como:

Comment [WU399]: Acho que a vírgula está no lugar errado, o correto seria "avance, projetos"

Comment [WU400]: Acho que essa vírgula deveria ser excluída. Na maioria dos casos não se usa vírgula após um "e"

- *Reunião de kick-off:* marca o início efetivo do projeto. É uma oportunidade dos participantes se conhecerem, se manifestarem quanto suas expectativas;
- *Reuniões para registro e acompanhamento de mudanças:* formalizam as possíveis mudanças no planejamento e execução do projeto, que certamente trará algum impacto no mesmo. A partir dessas reuniões, será gerado um documento formal, padronizado, que posteriormente poderá ser ou não avaliado e aprovado ou por um comitê executivo do projeto;

Comment [WU401]: Acho que o certo é "quanto às suas expectativas"

Comment [WU402]: Acho que o certo é "trarão" já que se refere a mudanças

Comment [WU403]: Acho que esse "ou" deve ser excluído pois está sobrando na frase.

É importante destacar que os gerentes de projetos devem procurar sempre estar bem

papel com eficiência e eficácia. Para alcançar tais objetivos alguns pontos devem ser levados em consideração pelos gerentes de projeto:

- Preparar de uma pauta;
- Realizar de intervalos em reuniões longas;
- Utilizar o máximo do tempo e o esforço de todos os envolvidos (os profissionais poderiam tratar do mesmo assunto (ou decisão) por outro meio de comunicação (por exemplo, e-mail, carta, teleconferência, etc);

Comment [WU404]: Esse "de" está sobrando. Ou tira ele ou coloca "Preparação" ao invés de "preparar"

Comment [WU405]: Idem ao 26

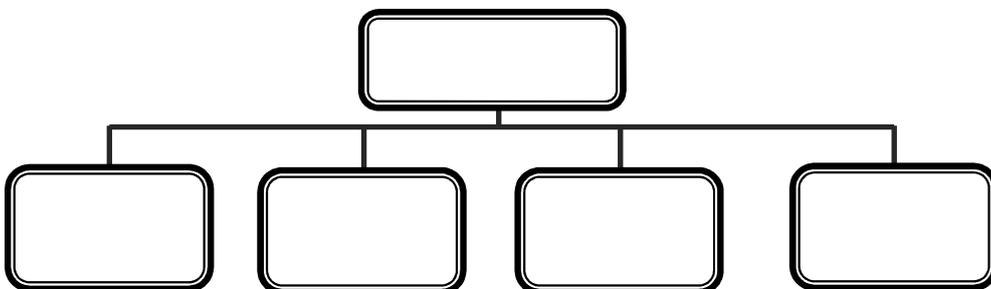
Comment [WU406]: Esse parêntese foi aberto e não foi fechado.

As comunicações do projeto sempre foram e continuarão sendo um ingrediente importante na fórmula para o seu sucesso. O PMBoK²¹ considera a área de conhecimento "comunicação", como sendo vital para projetos e seu sucesso. Por isso, a gerência da comunicação é considerada uma das áreas mais importantes na gerência de projetos, apesar de ser muitas vezes negligenciada

O gerenciamento das comunicações do projeto é a área de conhecimento que emprega os processos necessários para garantir a geração, coleta, distribuição, armazenamento, recuperação e destinação final das informações sobre o projeto de forma oportuna. O PMBOK considera como uma boa prática da gestão de projetos utilizar os ativos de processos organizacionais, que será explicado na Seção 14.2.1. Apesar de um projeto ser

Comment [WU407]: O correto é "serão explicados"

²¹ PMBOK é um *Guide to the Project Management Body of Knowledge* desenvolvido pelo *Project Management*



único e temporário, as informações geradas e o histórico de um projeto podem e devem ser considerados como base de dados para outro projeto semelhante. Para se ter uma boa gestão da comunicação em um projeto, segundo o PMBOK [PMBOK 2004], além do planejamento é preciso cuidar da distribuição das informações, do relatório de desempenho e gerenciar as partes interessadas como mostra a Figura 14.2.

Comment [WU408]: Corrigir a formatação dessa linha.

Figura 14.2. Visão geral do gerenciamento das comunicações do projeto [Adaptada de PMBOK 2004].

A seção anterior deste capítulo apresentou algumas questões relacionadas ao processo de comunicação em geral, sob o ponto de vista de sua utilização e importância no gerenciamento de projetos. O gerenciamento das comunicações em projetos estabelece, monitora e controla o fluxo de informações durante todo ciclo de vida dos projetos, sendo vital para o sucesso dos mesmos. Portanto, é de extrema importância que as comunicações em projetos sejam realizadas segundo processos organizados e disciplinados, capazes de gerar informações corretas e completas.

Comment [WU409]: Dar um espaçamento vertical antes e depois da figura e corrigi-la para que ela fique dentro das margens estabelecidas.

Os processos conforme mostrados na figura 2 se relacionam e interagem durante a condução do projeto, a descrição de cada um desses processos é feita por meio dos seguintes termos: *Entrada*: são representadas por documentos, planos, desenhos; *Ferramentas e Técnicas*: são aplicadas as entradas; *Saídas*: são representadas por documentos, resultados, produtos.

Comment [WU410]: Corrigir a referência para a figura "14.2"

O processo de planejamento das comunicações determina as necessidades informações e comunicações das partes interessadas PMBOK [PMBOK 2004]. O planejamento envolve a identificação e definição das seguintes informações: quais são as informações e quem as precisa; quando precisarão e com qual frequência; como será fornecida e por quem. Embora todos os projetos compartilhem a necessidade de comunicar informações, as necessidades das informações e os métodos de distribuição variam amplamente. Identificar as necessidades de informação dos interessados e determinar uma forma para atender a essas necessidades, é fator importante para o sucesso do projeto.

Comment [WU411]: O certo seria "necessidades de informações"

Comment [WU412]: O certo seria "quem precisa delas"

Em um número significativo de projetos a maior parte do planejamento da comunicação é feita como parte das fases iniciais do projeto. Entretanto, os resultados deste processo devem ser revistos regularmente durante o projeto e revisados se necessário para garantir aplicabilidade contínua. Esse planejamento é frequente e firmemente relacionado ao planejamento organizacional, visto que a estrutura organizacional do projeto terá um maior efeito nos requerimentos de comunicação. A

Comment [WU413]: Excluir o trema

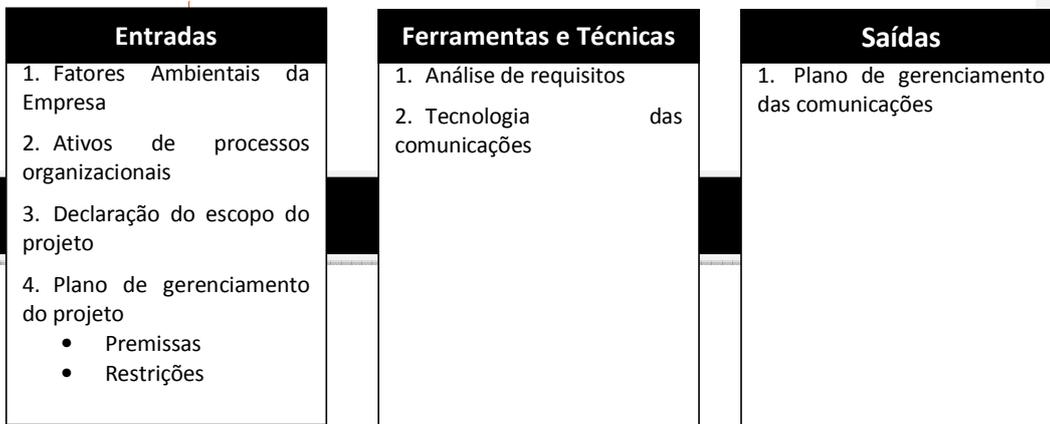


Figura 14.3. Planejamento das comunicações [Adaptada de PMBOK 2004].

1. Entradas para o Planejamento das comunicações:

Ao longo do desenvolvimento do planejamento das comunicações do projeto, os Fatores Ambientais da Empresa devem ser considerados todos e quaisquer sistemas e fatores ambientais da empresa que influenciam o sucesso da comunicação no projeto. Isso inclui, porém não se limita aos seguintes itens:

- Cultura e estrutura organizacional ou da empresa;
- Normas governamentais ou do setor;
- Recursos humanos existentes (por exemplo, com facilidades, habilidades no ato de comunicar-se);

Comment [WU415]: Idem comentário 31

Comment [WU416]: Acho que o número dessa seção deveria ser 14.2.1.1.

Comment [WU417]: Rever esta frase que contém trechos repetidos

Comment [WU418]: Retirar esse "ou"

Comment [WU419]: Idem 39

Comment [WU420]: Não entendi esse exemplo.

Ao longo do desenvolvimento do planejamento da comunicação e da documentação subsequente do projeto, todos ativos de processos organizacionais usados para influenciar o sucesso da comunicação do projeto podem ser obtidos a partir dos ativos de processos organizacionais. Todas e quaisquer organizações envolvidas no projeto podem ter políticas, procedimentos, planos e diretrizes formais e informais cujos efeitos devem ser considerados. Estes ativos também representam o aprendizado e o conhecimento das organizações obtidos de projetos anteriores; por exemplo, cronogramas terminados, dados de risco e dados de valor agregado. Tais ativos podem ser organizados de diversas formas, dependendo do tipo de setor,

Comment [WU421]: Idem ao 35

Comment [WU422]: Essa frase está sem sentido.

Comment [WU423]: O certo é "da organização obtido de projetos anteriores"

Comment [WU424]: Rever a pontuação a ser

organização para realizar o trabalho base de conhecimento corporativo da empresa para armazenar e recuperar informações. É importante ressaltar que as lições aprendidas, bem como as informações históricas podem fornecer decisões e resultados com base em projetos anteriores semelhantes relacionados a problemas de comunicações.

A declaração do escopo do projeto descreve detalhadamente as entregas do projeto e o trabalho necessário para criar as entregas. Esta declaração também fornece um entendimento sobre o escopo do projeto para todas as partes interessadas no projeto e descreve os principais objetivos do projeto. Além disso, permite que a equipe do projeto realize um planejamento mais detalhado, orienta o trabalho da equipe durante a execução de tarefas e fornece a linha de base para avaliar solicitações de mudanças ou trabalho adicional e verificar se estão contidos dentro ou fora dos limites do projeto. O nível e o grau de detalhamento com que a declaração do escopo do projeto define o trabalho que será realizado, bem como o que será excluído podem determinar a eficácia com que a equipe de gerenciamento de projetos poderá controlar o escopo global do projeto. O gerenciamento do escopo do projeto, por sua vez, pode determinar a eficácia com que a equipe planeja, gerencia e controla a execução do projeto. A declaração do escopo do projeto inclui, diretamente ou referencia outros documentos como:

Comment [WU425]: Não ficou claro quais são as duas categorias de processos organizacionais

Comment [WU426]: Verificar a repetição de palavras. Só a palavra "projeto" apareceu 5 vezes nesse trecho.

Comment [WU427]: O correto seria "a" ao invés de "e"

O plano de gerenciamento do projeto inclui as ações necessárias para definir, coordenar e integrar todos os planos auxiliares em um plano de gerenciamento do projeto. O conteúdo do plano de gerenciamento do projeto varia de acordo com a área de aplicação e a complexidade do projeto. Este plano define como o projeto é executado, monitorado, controlado e encerrado. Esse plano documenta o conjunto de saídas dos processos de planejamento do Grupo de processos de planejamento²² e inclui:

2. Ferramentas e Técnicas para o Planejamento das comunicações:

Comment [WU428]: Acho que o número dessa seção deveria ser 14.2.1.2.

- Logística de quantas pessoas serão envolvidas no projeto e em que locais;

Comment [WU429]: Acho que esse "de" deveria ser trocado por "e".

- Necessidades externas de informações (por exemplo, a comunicação com as contratadas ou com os meios de comunicação);

Comment [WU430]: Acho que você quis dizer "as empresas contratadas" seria bom explicitar a palavra empresas pra facilitar o entendimento.

- *A urgência da necessidade de informações:* O sucesso do projeto depende da pronta disponibilidade de informações atualizadas frequentemente ou relatórios por escrito emitidos regularmente seriam suficientes?

Comment [WU431]: Idem ao 35

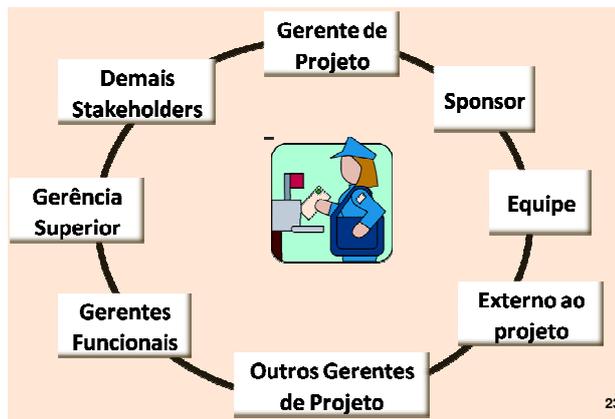
3. Saídas do Planejamento das comunicações:

Comment [WU432]: Acho que o número dessa seção deveria ser 14.2.1.3.

A distribuição das informações envolve disponibilizar as informações às partes interessadas no projeto no momento adequado. A distribuição das informações inclui implementar o plano de gerenciamento das comunicações, além de responder às solicitações de informações não previstas. Além disso, é imprescindível neste processo ter a clareza de quais e para quem as informações devem ser enviadas, conforme mostra as Figuras 14.4 e 14.5 respectivamente.

Comment [WU434]: Acho que ficaria melhor assim "...quais informações e para quem estas devem..."

Comment [WU435]: "mostram"



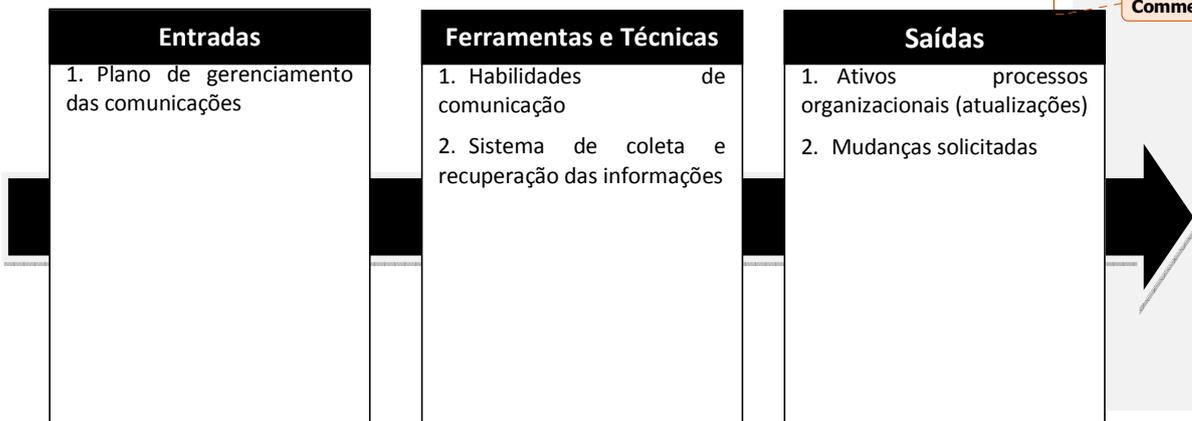
23

Figura 14.5. Para quem as informações que devem ser distribuídas [Adaptada de PERRELLI 2004].

A composição do processo de distribuição das informações pode ser visualizada na Figura 14.6.

Comment [WU436]: Dar um espaçamento vertical depois da figura, alterar a figura para escala de cinza e não colorida e retirar esse número "3" que está a direita dela

Comment [WU437]: Idem ao 31



1. Entradas para a Distribuição das informações:

Comment [WU438]: Acho que a numeração dessa seção deveria ser 14.2.2.1

2. Ferramentas e Técnicas para a Distribuição das informações

Comment [WU439]: Acho que a numeração dessa seção deveria ser 14.2.2.2

O sistema de coleta e recuperação das informações pode ocorrer por diversos meios, inclusive sistemas manuais de arquivamento, software de gerenciamento de projetos, bancos de dados eletrônicos, e sistemas que possibilitam o acesso à documentação técnica, como especificações de design, plano de testes e desenhos de engenharia.

Comment [WU440]: Acho que o certo é "possibilitem"

O processo de lições aprendidas se concentra na identificação dos sucessos e fracassos do projeto, incluindo sugestões para melhorar o desempenho futuro dos projetos. Durante o ciclo de vida do projeto a equipe e as principais partes interessadas identificam as lições aprendidas relacionadas aos aspectos técnicos, gerenciais do projeto. As lições aprendidas são compiladas, formalizadas e armazenadas durante o projeto. Além disso, é importante destacar alguns resultados específicos das lições aprendidas, como:

Comment [WU441]: Acho que ficaria melhor substituir a "," por um "e"

3. Saídas da Distribuição das informações

A atualização dos ativos processos organizacionais é composta basicamente:

Comment [WU442]: Acho que a numeração dessa seção deveria ser 14.2.2.3

Comment [WU443]: Acho que o certo é "ativos de processos"

Comment [WU444]: Acho que o certo é "composta basicamente de:"

- Apresentações do projeto. A equipe fornece informações, formal ou informalmente, a algumas ou a todas as partes interessadas no projeto. Essas informações são relevantes para as necessidades da audiência.
- *Feedback* das partes interessadas. As informações recebidas das partes interessadas relativas às operações do projeto podem ser distribuídas e usadas para modificar ou melhorar o desempenho futuro do projeto.
- Notificações das partes interessadas. É possível fornecer informações às partes interessadas sobre problemas resolvidos, mudanças aprovadas e andamento geral do projeto.

Comment [WU445]: Acho que seria ":" ao invés de ":",

Comment [WU446]: Idem ao 67

Comment [WU447]: Idem ao 67

As mudanças surgidas no processo de Distribuição das informações devem causar mudanças no plano de gerenciamento do projeto e no plano de gerenciamento das comunicações. As mudanças solicitadas (adições, modificações, revisões) no plano de gerenciamento do projeto e nos seus planos auxiliares são revisadas e a destinação é gerenciada pelo processo Controle integrado de mudanças²⁴.

O processo de relatório de desempenho envolve a coleta de todos os dados de linha de base e a distribuição das informações sobre o desempenho às partes interessadas. Em geral, o relatório de desempenho contém informações, incluindo o modo como os recursos estão sendo utilizados para atingir os objetivos do projeto.

Comment [WU448]: Acho que o certo aqui seria "da" ao invés de "de"

Comment [WU449]: Idem ao 31

Entradas

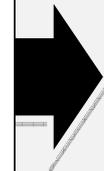
1. Informações sobre o desempenho do trabalho
2. Medições de desempenho
3. Previsão de término
4. Medições de controle de

Ferramentas e Técnicas

1. Ferramentas de apresentação das informações
2. Coleta e compilação das informações sobre o

Saídas

1. Relatórios de desempenho
2. Previsões
3. Mudanças solicitadas



1. Entradas para o Relatório de desempenho

As informações sobre o desempenho do trabalho tratam a situação atual das entregas. Já referente ao que foi realizado no projeto, as informações são coletadas como parte da execução e alimentadas no processo Relatório de desempenho. As informações sobre o andamento das atividades que estão sendo executadas para realizar o trabalho do projeto são coletadas rotineiramente como parte da execução do plano de gerenciamento do projeto. Essas informações incluem, mas não estão limitadas a:

Comment [WU450]: Acho que a numeração dessa seção deveria ser 14.2.3.1

Comment [WU451]: Acho que o certo seria "alimentação do"

Tabela 14.2. Valores-chave da técnica do valor agregado.

		Custo orçado do trabalho agendado a ser terminado em uma atividade ou o componente da EAP ²⁵ até um determinado momento.
		Custo total incorrido na realização do trabalho na atividade

Comment [WU452]: Colocar a legenda na mesma página que a tabela

		do cronograma ou no componente da EAP durante um determinado período de tempo. Este CR deve corresponder em definição e em cobertura a tudo o que foi orçado para o VP e o VA (por exemplo, somente horas diretas, custos diretos ou todos os custos, inclusive custos indiretos).
		A VC é igual ao valor agregado (VA) menos o valor planejado (VP). A variação de prazos será no final igual a zero quando o projeto for terminado, devido todos os valores planejados terem sido agregados.

Comment [WU453]: Acho que o certo aqui seria "VP"

A previsão de término inclui a realização de estimativas a respeito das condições futuras do projeto com base nas informações e nos conhecimentos disponíveis no momento da previsão. As previsões são geradas, atualizadas e refeitas com base nas informações sobre o desempenho do trabalho fornecidas conforme o projeto é executado e desenvolvido. Essas informações sobre o desempenho do trabalho se referem ao desempenho passado do projeto

para terminar atividades do cronograma, o que é denominado ENT. As técnicas de previsão também ajudam a determinar a EPT, que é a estimativa para terminar o trabalho restante de uma atividade do cronograma, um pacote de trabalho ou uma conta de controle. Embora a técnica do valor agregado de determinação da ENT e da EPT seja rápida e automática, ela não é tão valiosa ou exata quanto uma previsão manual do trabalho restante a ser realizado pela equipe do projeto. Os valores de ENT e EPT são calculados ou relatados pela organização executora, em seguida são documentados e comunicados às partes interessadas.

As medições de controle da qualidade são os resultados das atividades de controle da qualidade fornecidos como *feedback* para o processo de Garantia da Qualidade (GQ)²⁶ para uso na reavaliação e na análise dos processos e padrões de qualidade da organização executora.

O plano de gerenciamento do projeto fornece informações sobre a linha de base. A linha de base da medição de desempenho trata-se de um plano aprovado para o trabalho do projeto em relação ao qual a execução do projeto é comparada, sendo medidos os desvios para o controle gerencial. A linha de base da medição de desempenho normalmente integra os parâmetros de escopo, cronograma e custo de um projeto, mas pode também incluir parâmetros técnicos e de qualidade.

Comment [WU454]: Excluir esse espaçamento vertical

2. Ferramentas e Técnicas para o relatório de desempenho

As ferramentas de apresentação das informações incluem os pacotes de software que por sua vez contêm análise de planilhas, relatórios de tabelas, apresentações ou capacidades gráficas podem ser usados para criar imagens de qualidade para a apresentação dos dados de desempenho do projeto.

Comment [WU455]: Acho que a numeração dessa seção deveria ser 14.2.3.2

Comment [WU456]: Acho que o certo é "gráficas que podem"

3. Saídas do Relatório de desempenho

Os relatórios de desempenho organizam e sintetizam as informações coletadas e apresentam os resultados de qualquer análise comparados com a linha de base da medição de desempenho. Os relatórios devem fornecer informações sobre o progresso e o andamento e o nível de detalhes exigido pelas diversas partes interessadas, conforme documentado no plano de gerenciamento das comunicações.

Comment [WU457]: Acho que a numeração dessa seção deveria ser 14.2.3.3

Comment [WU458]: Acho que deveria ser uma “.” ao invés do “e”

- **PMO:** Se existir na organização executora, o PMO poderá ser uma parte interessada se tiver responsabilidade direta ou indireta pelo resultado do projeto.

Comment [WU459]: Acho que seria interessante explicitar o que é um PMO para melhorar o entendimento do capítulo.

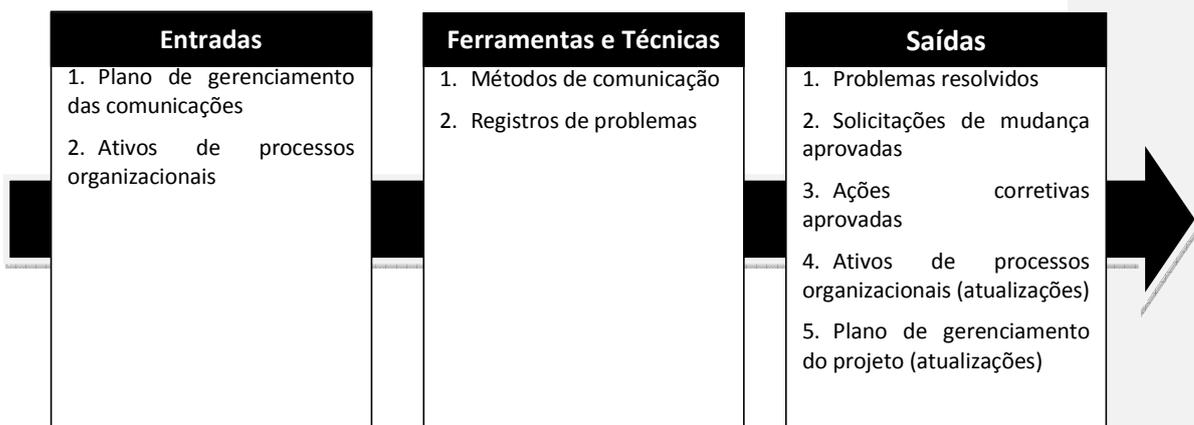


Figura 14.11. Gerenciar as partes interessadas [Adaptado de PMBOK 2004].

Comment [WU460]: Idem ao 31

1. Entradas para Gerenciar as partes interessadas

Os requisitos e expectativas das partes interessadas propiciam um entendimento das metas, objetivos e nível de comunicação das partes interessadas durante o projeto. As necessidades e expectativas são identificadas, analisadas e documentadas no plano de gerenciamento das comunicações (Seção 2.1.3).

Comment [WU461]: Acho que a numeração dessa seção deveria ser 14.2.4.1

Comment [WU462]: Verificar qual o número correto da seção a ser referenciada

3. Saídas de Gerenciar as partes interessadas

Comment [WU464]: Acho que a numeração dessa seção deveria ser 14.2.4.3

processos, projetos e atividades de âmbito empresarial. A comunicação deve ser objetiva e clara, respeitando os limites estipulados por agentes definem o negócio, seja de um projeto ou organização. Tendo em vista que as organizações que apresentam uma comunicação mais eficaz garantem seu espaço no mercado, conseqüentemente apresentam resultados mais satisfatórios, assim, garantindo a satisfação dos clientes. Por este motivo, tem surgido uma grande demanda ao incentivo de pesquisas que levem em consideração à procura por formas de melhoria das práticas do processo de comunicação.

Este capítulo procurou introduzir ao leitor a importância e boas práticas que diz respeito à comunicação, apresentando o processo de comunicação em Organizações e projetos, sendo destacadas as principais barreiras, estratégias e quão é necessária a eficácia da comunicação. O gerenciamento das comunicações em projetos também foi destacado, pois estabelece, monitora e controla o fluxo de informações durante todo ciclo de vida dos projetos, sendo vital para o sucesso dos mesmos. Neste gerenciamento também foram apresentados um conjunto processos, os quais como mais uma tentativa de atingir os objetivos desejados, de maneira natural e transparente, minimizando os riscos, conflitos e frustrações que possam vir ocorrer ao longo do projeto.

1. **Comunicação Empresarial:**

A área de comunicação empresarial tem um papel importante na "administração de percepção" e na leitura do ambiente social da organização. Nessa perspectiva várias pesquisas, propostas, ferramentas vem contribuindo para a análise dos planos de negócios da organização, identificando problemas e oportunidades no campo da comunicação para garantir o sucesso da empresa no mercado.

2. **Comunicação em Desenvolvimento Distribuído de Software (DDS):**

Comment [WU465]: Acho que o certo seria "agents que definem"

Comment [WU466]: Excluir o trema

Comment [WU467]: Acho que ficaria melhor se esse texto fosse substituído por "referentes ao tema"

Comment [WU468]: Acho que o certo seria "conjunto de processos"

Comment [WU469]: Acho que esse "vir" deve ser retirado do texto

Comment [WU470]: Acho que a numeração deveria ser 14.4.1

Comment [WU471]: Acho que o certo seria "e" ao invés da " ,"

Comment [WU472]: Acho que a numeração deveria ser 14.4.2

2. Para ampliar o conhecimento sobre o Gerenciamento da Comunicação é recomendada a leitura do livro Gerenciamento da comunicação em projetos, Chaves, L; Neto, F.; Pech, G.; Carneiro, M., 2006, o qual faz parte da Série Gerenciamento de Projetos das Publicações Fundação Getúlio Vargas. Este livro visa oferecer a estudantes, gestores e técnicos uma base didáticas para que estes possam aplicar o processo de comunicação em uma maneira eficaz em projetos de software.
3. Para um melhor conhecimento sobre a Comunicação Empresarial é sugerida a leitura das revistas: Comunicação Empresarial da Associação Brasileira de Comunicação Empresarial (Aberje) e a Comunicação Corporativa do Valor Setorial, as quais retratam a comunicação empresarial sobre diversos aspectos relacionados a atualidade. Tais revistas podem ser visualizadas, respectivamente nos seguintes links: <http://www2.ideavalley.com.br/aberje/flip/> e <http://208.96.41.18/valoreconomico/home.aspx?pub=27&edicao=1>.
4. Para ampliar o entendimento sobre Comunicação em Desenvolvimento Distribuído de Software é sugerida a leitura da dissertação Uma Proposta de Boas Práticas do Processo de Comunicação para Projetos de Desenvolvimento Distribuído de Software, Junior, L. Nesta dissertação é mostrada práticas da comunicação em DDS, a partir de uma experiência prática de empresas.

Comment [WU473]: Acho que o certo é "didática"

Comment [WU474]: Acho que o certo é "de"

Comment [WU475]: Seria interessante colocar qual foi a data do ultimo acesso, pois estas fonts podem sair do ar futuramente.

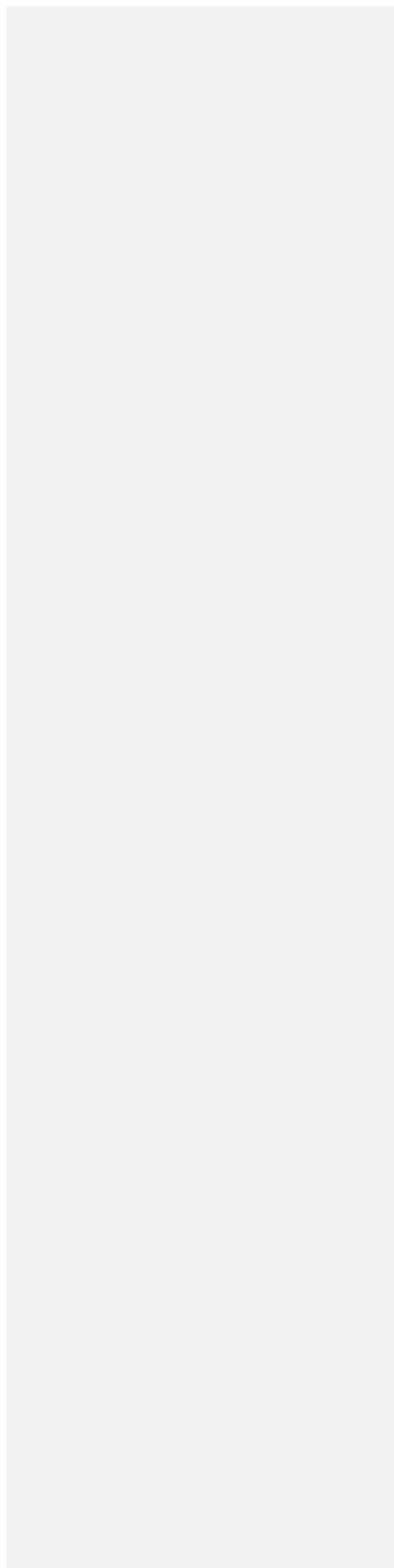
Comment [WU476]: Acho que o correto seria "são"

13. Cite quais informações e para quem devem ser distribuídas.

Comment [WU477]: Não ficou claro o objetivo dessa pergunta, melhor reformular ou excluir.

Chaves, L. Gerência de Comunicação em Projetos. Disponível em:
http://www.docstoc.com/docs/4565064/Slides-Gerencia-de-Comunica%C3%A7%C3%

[estrategia-da-comunicacao-na-implementacao-do-planejamento-estrategico-42653.html](#).
Acessado em: set. 2009.



Vargas, R. (1999). Gerenciamento de Projetos - Estabelecendo diferenciais competitivos, 6ª edição, Editora Braspor.

Comment [WU478]: Acho que o nome da editora é "Brasport"

Verzuh, E. (2000). MBA compacto, Gestão de Projetos. 3. ed. Rio de Janeiro: Campus.398 p.

15.1. IMPORTÂNCIA DA MEDIÇÃO ERROR! BOOKMARK NOT DEFINED.

15.2. O QUE SÃO MÉTRICAS ERROR! BOOKMARK NOT DEFINED.

15.3. MEDIÇÃO DE SOFTWARE ERROR! BOOKMARK NOT DEFINED.

15.3.1. MÉTRICAS TÉCNICAS ERROR! BOOKMARK NOT DEFINED.

15.3.2. MÉTRICAS DE QUALIDADE ERROR! BOOKMARK NOT DEFINED.

15.3.3. MÉTRICAS DE PRODUTIVIDADE ERROR! BOOKMARK NOT DEFINED.

15.3.4. MÉTRICAS ORIENTADAS AO TAMANHO ERROR! BOOKMARK NOT DEFINED.

15.3.5. MÉTRICAS ORIENTADAS À FUNÇÃO ERROR! BOOKMARK NOT DEFINED.

15.3.6. MÉTRICAS ORIENTADAS À SERES HUMANOS ERROR! BOOKMARK NOT DEFINED.

15.4. BOAS PRÁTICAS NA IMPLANTAÇÃO DE PROGRAMAS DE MEDIÇÃO NAS ORGANIZAÇÕES
ERROR! BOOKMARK NOT DEFINED.

15.5. ESTIMATIVAS DE SOFTWARE ERROR! BOOKMARK NOT DEFINED.

15.6. POR QUE É DIFÍCIL ESTIMAR ERROR! BOOKMARK NOT DEFINED.

15.7. TÉCNICAS DE ESTIMATIVA ERROR! BOOKMARK NOT DEFINED.

15.7.1. FPA ERROR! BOOKMARK NOT DEFINED.

15.7.2. PONTOS DE CASO DE USO ERROR! BOOKMARK NOT DEFINED.

15.7.3. COCOMO ERROR! BOOKMARK NOT DEFINED.

[15.9. TÓPICOS DE PESQUISA](#) ERROR! BOOKMARK NOT DEFINED.

Comment [A479]: Primeiro é Tópicos de Pesquisa e depois Sugestões de Leitura.

[15.10. EXERCÍCIOS](#) ERROR! BOOKMARK NOT DEFINED.

[REFERÊNCIAS](#) ERROR! BOOKMARK NOT DEFINED.

Neste capítulo são considerados conceitos fundamentais acerca da medição e estimativas de software. O capítulo começa trazendo uma visão geral sobre medição de software, com a motivação para se realizar este tipo de atividade, conceitos básicos e métricas de software. A discussão, em seguida, passa a ser a respeito de modelos de processo de medição de software.

Comment [A480]: Seria interessante neste espaço mostrar o objetivo do capítulo, antes de abordar o que o capítulo apresenta.

Além do que já foi dito no parágrafo anterior, boas práticas na implantação de

Comment [A481]: Acho que ficaria melhor deste jeito:
"O capítulo aborda uma visão geral sobre medição de software, a sua importância, conceitos básicos,

momento de levar à empresa a instituição de um programa de medição. Pontos de Caso de Uso, COCOMO, entre outras técnicas para estimativas em projetos de software, formam a última parte do capítulo.

É fato que a indústria de software continua, até hoje, lidando com projetos de software mal sucedidos. Uma pesquisa do [The Standish Group 2009] apontou que mais projetos estão falhando e menos estão tendo sucesso. E uma das causas que pode ser apontada para este problema é uma gestão não tão bem planejada e executada dos projetos de software. Uma boa maneira de realizar uma gestão de projeto de software com um pouco mais de garantias de que o trabalho realizado não será um fracasso total, é executando, entre outras coisas, atividades de medição e estimativas. Afinal de contas, e em concordância com as palavras de [Tom DeMarco 1982], não se pode controlar o que não se pode medir. Quando o software é medido, é feito, entre outras funções, de dar ao gerente do projeto de software valores reais para que possa enxergar o projeto de uma maneira quantificada, apoiando a tomada de decisões e, subsídios para que ele possa realizar estimativas mais próximas da futura realidade final dos projetos que estão por vir. Além deste fator, o panorama encontrado em muitos projetos de software é o mesmo, e não é animador, apontando para a necessidade de medições. [Fernandes 1995] enumera 12 situações comuns a quem desenvolve software e que não contribuem em nada para uma gestão mais efetiva de projetos deste tipo de produto, as quais:

1. Estimativas de prazos, custos, recursos e esforço são realizados com base no julgamento pessoal do gerente de projeto.

7. Os fatores que impactam a produtividade e a qualidade não são determinados.

9. Os custos de não conformidade ou da má qualidade não são medidos.

Apesar da implantação de um programa de métricas na empresa ser uma atividade que represente mais trabalho e traga alguns custos imediatos, fora o fato de dar a impressão inicial de que tudo aquilo não está sendo útil, os ganhos futuros com a formação de uma base de dados composta por métricas de projetos de vários anos farão deste repositório uma "voz" muito precisa durante a estimativa de prazos, custos, esforço e recursos em outros projetos,

Comment [A482]: Acho que este parágrafo poderia ser continuidade do anterior. Acho que ficaria melhor assim:
"Além da visão geral, também é realizada uma abordagem sobre as boas práticas na implantação de programas de medição nas organizações e as possíveis técnicas para estimativas em projetos de software, como por exemplo: Pontos de Caso de Uso e COCOMO."

Comment [A483]: Acho que ficaria melhor assim:
Segundo a pesquisa de [The Standish Group 2009] foi detectado que existe um número mais significativo de falhas dos projetos do que o sucesso nos mesmos.

Comment [A484]: Uma das causas que reflete este problema é uma gestão mal planejada e organizada

Comment [A485]: Acho que esse paragrafo pode ser unido com o anterior e poderia ficar assim:
"Diante desse contexto, é importante destacar que uma boa maneira de realizar gestão de projeto de software é aplicando atividades de medição e estimativa, a fim de garantir que o trabalho desenvolvido não será totalmente um fracasso. Portanto, não se pode controlar o que não se pode medir [Tom DeMarco 1982]".

Comment [A486]: Acho que poderia ficar assim:
"Quando o software é construído, e avaliado, tem como finalidade fornecer ao gerente do projeto de software valores reais para que possa enxergar o projeto de uma maneira quantificada, apoiando a tomada de decisões e, subsídios para seja possível a realização de estimativas mais próximas da realidade final dos futuros projetos."

Comment [A487]: Poderia ficar assim:
"Além deste fator, o panorama encontrado em diversos projetos de software é o mesmo. Portanto, panorama não é animador, já que aponta para a necessidade de medições. De acordo com Fernandes [Fernandes 1995], existem doze situações comuns referente aos desenvolvedores de software, as quais não contribuem positivamente para uma gestão mais efetiva de projetos deste tipo de produto, tais como:"

Comment [A488]: Ao invés de números acho que fica melhor colocar marcadores.

Comment [A489]: Na produtividade e na qualidade.

Comment [A490]: péssima

Comment [A491]: acho que fica melhor representar

Comment [A492]: conceba

Comment [A493]: oferecer

Comment [A494]: seria melhor mudar essa expressão, pois ficou meio estranha no texto.

Comment [A495]: proporcionando

Diferentemente das outras áreas da engenharia, onde a medição é algo que rege o trabalho realizado dentro das atividades destas ciências, na engenharia de software a medição ainda se encaminha para uma tentativa de firmamento dentro dos processos que se valem desta área do conhecimento. O que acontece aqui, na verdade, é que o fato de medir software e os processos para o desenvolvimento de tais produtos parece, a muitos, algo tão abstrato e subjetivo. No entanto, medidas de software e métricas têm sido derivadas ao longo do tempo para que medições do produto sejam feitas.

No geral, a importância das métricas de software relaciona-se ao fato de darem aos engenheiros de software um modo sistemático de avaliar a qualidade do produto que é desenvolvido e também o processo utilizado para tal com base em um conjunto de regras claramente definidas, permitindo que os gestores do projeto tenham um entendimento imediato do que está sendo feito, e não posteriormente. Tudo isso faz com que eles possam descobrir problemas no decorrer do projeto antes que estes se transformem em algo muito mais difícil de ser resolvido mais tarde. Em suma, o processo de software é medido num esforço para melhorá-lo, ao passo que o produto é medido num esforço para aumentar sua qualidade [Pressman 1995].

15.2. O que são Métricas

Existem alguns conceitos comuns em discussões sobre medições de software e que valem a pena serem discutidos a fim de trazer uma maior clareza sobre o assunto. Primeiramente, a definição do termo métrica. Pode-se dizer, de certa forma, que não há nenhuma definição aceita como a mais correta para o termo. Alguns profissionais usam o termo métrica intercambiadamente com o termo medição. Já outros fazem a distinção entre medição e métrica, onde métrica indica uma medição e um modelo ou teoria baseados nela [Shepperd e Ince 1993].

A definição dada no parágrafo anterior para o termo métrica parece não esclarecer muito acerca desta. No entanto, [Shepperd e Ince 1993] dão um bom entendimento do conceito de métrica ao afirmarem que esta, em engenharia de software, é nada mais e nada menos, do que aquilo que transmite uma medição de um produto ou processo de software.

Já dá para se ter, a partir do que foi exposto, uma boa noção acerca da definição do termo métrica. Mas tomando como referência a própria definição citada por [Shepperd e Ince 1993], vem, de antemão, o questionamento sobre o que vem a ser medição e também, diretamente relacionado, o seu substantivo correlato: medida. Pode-se entender medição como o ato de obter valores de uma característica ou atributo de uma coisa ou entidade qualquer. Ou seja, tomando como exemplo uma pessoa. A respeito da entidade pessoa existe um conjunto de características ou atributos os quais são passíveis de receberem valores, tal qual a altura, o peso e etc. O peso de uma pessoa pode ser 85 e sua altura 180. Referente a esses valores obtidos, costuma-se atribuir unidades de medida que serão responsáveis por dar a noção de quantidade de um valor qualquer e que servem de base para comparações com outros valores com mesma unidade de medida. No caso, poderíamos ter peso igual a 85 kg e altura igual a 180 cm.

O mesmo acontece com o software. Enxergando o software e o processo usado para o seu desenvolvimento como entidades, existem atributos pertinentes a estas duas coisas que podem ser medidos. [Pressman 2006] exemplifica o abordado até agora dizendo que quando um único ponto de dados foi coletado (por exemplo, o número de erros descoberto em um único componente de software), uma medida foi estabelecida, e que uma métrica de software é aquela que relaciona medidas individuais de algum modo (por exemplo, o número médio de

Comment [A497]: Poderia retirar o dentro e ficar somente "... nas atividades..."

Comment [A498]: Acho que poderia ficar assim: "... na engenharia de software, por exemplo,"

Comment [A499]: Poderia retirar essa expressão e começar com o "Na verdade..."

Comment [A500]: Pode ser retirado

Comment [A501]: Feitas é no sentido de construídas ou de Realizadas?

Comment [A502]: Em geral fica melhor

Comment [A503]: Oferecerem

Comment [A504]: Acho q ficaria melhor assim: "...do produto, o qual é desenvolvido e também o processo é utilizado com base em um conjunto de regras claramente definidas, permitindo que os gestores do projeto tenham um entendimento imediato do que está sendo realizado."

Comment [A505]: Melhor mudar essa expressão.

Comment [A506]: Acho q fica melhor assim: "...possam descobrir problemas no decorrer do projeto o mais rápido possível, antes que estes se transformem em problemas mais difíceis de serem resolvidos."

Comment [A507]: Definição de métricas

Comment [A508]:
Acho q ficaria melhor assim:
"Dentre esses conceitos, pode ser destacado a definição de métrica, sendo que não existe nenhum conceito mais correto para este termo. Portanto, alguns profissionais usam o termo métrica intercambiadamente com o termo medição. No entanto, outros fazem a distinção entre medição e métrica, onde métrica indica uma medição e um modelo ou teoria baseados nela [Shepperd e Ince 1993]."

Comment [A509]: Acho que pode juntar com o paragrafo anterior e ficaria assim:
Apesa r das definições expostas, o conceito do termo métrica ainda não ficou claro. No entanto, [Shepperd e Ince 1993] oferecem um bom entendimento do conceito ao afirmarem que esta, em engenharia de software, é aquilo que transmite uma medição de um produto ou processo de software.

Comment [A510]: Acho que ficaria melhor assim:
De acordo com os conceitos expostos, ficou clara a definição do termo métrica.

Comment [A511]: Acho que ficaria melhor assim:
No entanto, ao tomar como referência a própria definição de Shepperd & Ince [Shepperd & Ince 1993] (... [1]

Comment [A512]: Acho que ficaria melhor assim: (... [2]

(... [3]

Visto também como um conceito importante em discussões sobre medição de software é o termo indicador. O objetivo do engenheiro de software com a coleta de medidas do processo e do produto final é, a partir das medidas que ele tem em mãos, originar métricas de tal modo que chegue a indicadores os quais serão úteis para os gerentes de projeto ou à alta administração tomar as melhores decisões quanto ao projeto em vigor e estimar bem com relação a outros que estão por vir. Sendo assim, um indicador pode ser definido como uma métrica ou um conjunto de métricas que fornecem profundidade na visão do processo de software, do projeto ou do produto, permitindo assim que os engenheiros de software/gerente do projeto possam tornar o software ou o processo para seu desenvolvimento melhores, tudo isto a partir das conclusões tiradas dos indicadores [Pressman 2006].

No mundo físico, as medições podem ser divididas em duas categorias: medidas diretas e medidas indiretas. Tomando como exemplo um pneu, pode-se considerar como uma medida direta deste, a sua circunferência. E como uma medida indireta do pneu pode-se considerar a sua qualidade, medida, por exemplo, através de testes de resistência [Caramoni et al. 2008].

Com o software, as coisas funcionam de forma idêntica.

Entre as medidas diretas do processo de engenharia de software estão o custo e o esforço aplicados para tal. No software, linhas de código escritas, velocidade de execução, tamanho da memória e defeitos registrados ao longo de um determinado espaço de tempo são medidas diretas do produto. Funcionalidade, qualidade, complexidade, eficiência, confiabilidade, etc., são atributos de medidas indiretas do software [Pressman 2006].

Segundo [Pressman 1995], as métricas de software podem ser divididas em mais categorias próprias. O autor divide as métricas de software nas seguintes categorias:

Nas próximas seções cada uma das categorias de métricas de software é estudada.

15.3.1 Métricas Técnicas

A qualidade do software pode ser medida tanto durante o seu desenvolvimento quanto depois que o produto tiver sido implementado. As métricas de qualidade derivadas durante o desenvolvimento do produto formam uma base quantitativa para a tomada de decisões referentes ao projeto e aos testes que serão feitos com o software. Concentram-se

Comment [A514]: Não tem mt haver com o que foi abordado anteriormente.

Comment [A515]: Colocar virgule depois de indicadores

Comment [A516]: úteis para os gerentes de projeto ou para a alta administração ao tomar as melhores decisões quanto ao projeto em vigor e estimar da melhor forma os projetos futuros.

Comment [A517]: Colocar virgule depois do permitindo assim.

Comment [A518]: Retirar a barra e colocar o "e".

Comment [A519]: Retirar o tudo isto.

Comment [A520]: Acho que fica melhor "retiradas".

Comment [A521]: Acho que ficaria melhor assim:
Tais medições também funcionam de forma

Comment [A522]: Excluir esse pedaço.

Comment [A523]: É melhor substituir por entre outros.

Comment [A524]: Substituir por: Segundo Pressman [Pressman 1995]

Comment [A525]: Substituir por : sera apresentada.

Comment [A526]: Vais acrescentar mais assunto neste tópico?

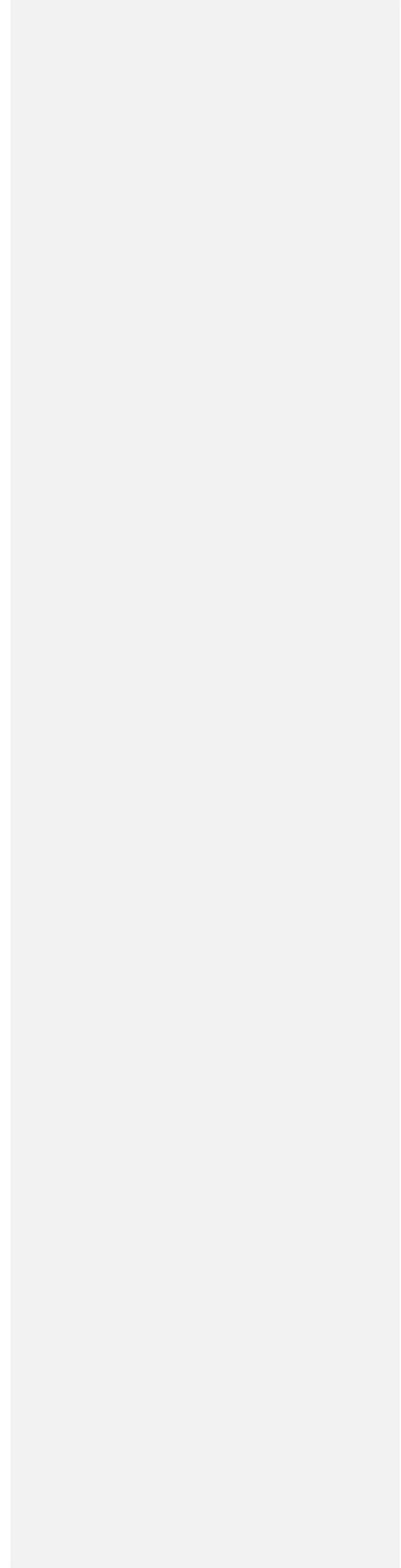
projeto sobre a efetividade do processo de engenharia de software aplicada ao projeto. Especial atenção é dada ao número de defeitos descobertos e à manutenibilidade do sistema [Boaventura 2001]. ...

Comment [A528]: Excluir essas palavras.

Comment [A529]: Acho que é melhor assim. Uma atenção em especial

Guarizzo, Karina, (2008) “Métricas de Software”, <http://bibdig.poliseducacional.com.br/document/?view=184>.

Comment [A530]: Organizar as referências, de acordo com o padrão estabelecido para o livro.



17 GESTÃO DE PROGRAMAS 109

17.1 PROGRAMAS..... 109

17.2	GERENCIAMENTO DE PROGRAMAS	111
17.2.1	RELAÇÃO ENTRE GERENCIAMENTO DO PROGRAMA E GERENCIAMENTO DO PROJETO	111
17.2.2	TEMAS DO GERENCIAMENTO DE PROGRAMA.....	112
17.2.2.1	GERENCIAMENTO DE BENEFÍCIOS	112
17.2.2.2	GERENCIAMENTO DE STAKEHOLDERS	113
17.2.2.3	GOVERNANÇA	114
17.2.3	CICLO DE VIDA DO PROGRAMA	115
17.2.3.1	FASE 1: SET UP PRÉ-PROGRAMA	116
17.2.3.2	FASE 2: SET UP PROGRAMA.....	117
17.2.3.3	FASE 3: ESTABELECEER ESTRUTURA DE GESTÃO DO PROGRAMA	118
17.2.3.4	FASE 4: BENEFÍCIOS INCREMENTAIS	118
17.2.3.5	FASE 5: ENCERRAMENTO	119
17.3	PROCESSOS DO GERENCIAMENTO DE PROGRAMA	119
17.3.1	GRUPO PROCESSOS DE INICIAÇÃO	120
17.3.2	GRUPO PROCESSOS DE PLANEJAMENTO	121
17.3.3	GRUPO PROCESSOS DE EXECUÇÃO.....	123
17.3.4	GRUPO PROCESSOS DE MONITORAMENTO E CONTROLE.....	124
17.3.5	GRUPO PROCESSOS DE ENCERRAMENTO.....	125

17.4 TÓPICOS DE PESQUISA..... 126

17.5 SUGESTÕES DE LEITURA..... 126

17.6 EXERCÍCIOS 126

17.7 REFERÊNCIAS..... 127

Comment [A531]: Criar um índice para figuras e tabelas.

17

Gestão de Programas

Capítulo

Comment [T532]: Deve estar em Times New Roman 18 e negrito.

Comment [T533]: Deve ser tamanho 20

Programas

Segundo o PMI, um Programa pode ser conceituado como um conjunto de Projetos **que** se relacionam de alguma forma e **que**, se tratados em modelo de Programa, tendem a trazer benefícios globais, onde projetos gerenciados de forma individual não obteriam. Um conceito **que** é importante saber, **que**, Programas também podem incluir o tratamento de operações oriundas do contexto de um Programa até **que** o mesmo se encerre.

Programas podem incluir elementos de trabalho relacionados fora do escopo dos projetos distintos no programa. Um grupo de projetos relacionados e gerenciados de modo coordenado para a obtenção de benefícios e controle **que** não estariam disponíveis se eles fossem gerenciados individualmente.

Segundo o IPMA (International Project Management Association), um programa consiste em um conjunto de propostas específicas e inter-relacionadas (projetos ou outras tarefas adicionais), que em conjunto convergem para uma finalidade comum, segundo uma determinada estratégia abrangente.

Portanto, as organizações ao implantarem programas, devem garantir que: estejam

Comment [A534]: Corrigir a sequência do tópicos com o capítulo.

Comment [T535]: Ajustar a numeração correta do capítulo.

Comment [T536]: Deve ser tamanho 14 e times New Roman e em negrito.

Comment [T537]: Tenta diminuir o uso da palavra "que" nesse parágrafo, porque ta muito repetido, tem 5 só aqui.

Comment [T538]: É uma definição de Programas? Ela está no parágrafo anterior com pequenas alterações nas palavras, não? Se realmente tiver ela aqui também, falta algum complemento antes pra ligar a Programas, porque dessa forma parece meio solta.

Comment [T539]: Tirar dúvida se é pra escrever palavras estrangeiras em itálico ou colocar entre parênteses.

Comment [T540]: Acho que esse verbo deveria

projetos; e os melhores recursos **devem** estar disponíveis. Ligação direta com o nível de maturidade pode chegar ao sucesso, o qual é gerido através:

Comment [T541]: A mesma coisa do comentário anterior, deveria ser "devam"

E por fim, as organizações terão que alcançar suas metas e objetivos ligados diretamente ao planejamento estratégico da organização; idealizá-las como o nível mais alto ao qual o trabalho está direcionado, diretamente ligado aos benefícios que trará para a organização. Esses benefícios são resultados que **provêm** utilidade aos interessados no Programa.

Comment [T542]: Se for metas e objetivos, a concordância deve ser dessa forma: "idealizá-los"

Comment [T543]: Quem trará? Se for metas e objetivos, a concordância correta deve ser "trarão"

Comment [T544]: Não tem mais esse acento, segundo o novo acordo ortográfico.

Comment [T545]: Está com a fonte diferente do texto. Formatar para o times New Roman 12

Comment [T546]: A Figura não pode ser colorida. Deve estar em tons de cinza.

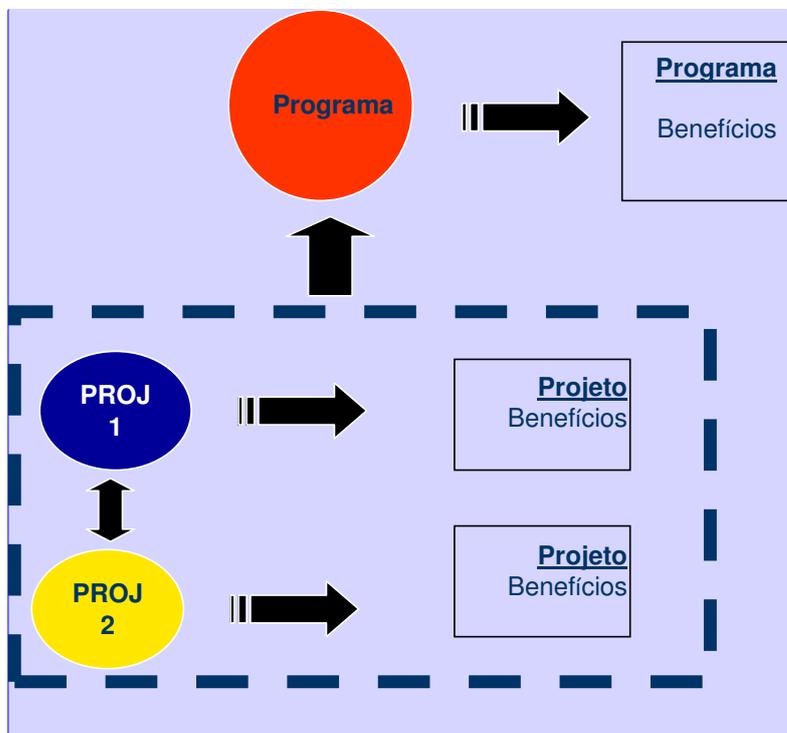


Figura 17.1 – Visão de Programa. [ADONAI 2008]

Comment [T547]: Seria interessante contextualizar a figura no texto do Capítulo e falar sobre ela. Dessa forma como se encontra, está solta no Capítulo.

Comment [T548]: Não deve estar em negrito, só a primeira parte com a numeração da figura. E deve ser tamanho 12. O mesmo vale para todas as outras figuras e tabelas.

Exemplos de programas:

Comment [T549]: Todos os três exemplos são gestão de programas, seria melhor colocar esse nome Gestão de Programas em cima, do que repeti-lo três vezes, num é? Esteticamente você não acha que ficou um pouco estranho a do meio sem Figura para ilustrar e as outras com figuras? Acho que deveria padronizar, ou todos sem figura, ou com figura..





Comment [T551]: Não pode ter figuras coloridas

Relação entre Gerenciamento do Programa e Gerenciamento do Projeto

Após introduzir a teoria de Gerenciamento do Programa, podem surgir dúvidas para conseguir separar onde entra o conceito de Gerenciamento de Programas e o conceito de Gerenciamento do Projeto. Portanto, abaixo, estaremos descrevendo um paralelo entre esse dois gerenciamentos.

Comment [T552]: Não ficaria melhor a preposição DE: Gerenciamento de Programa e Gerenciamento de Projeto?

Comment [T553]: A fonte não está de acordo com o padrão. Deve ser Times New Roman 13 negrito

Comment [T554]: O texto deve ser impessoal, segundo os professores Hermano e Alexandre.

Comment [T555]: Fonte diferente do padrão.

--	--

mudanças	mudanças e promovê-las
Gerentes de projetos lideram técnicos, especialistas etc.	
	Monitoram os projetos e a operação continuada através da estrutura de governança
	Re-planejado constantemente

Comment [T556]: Muda para : entre outros.

Comment [T557]: Como aqui não tem o sujeito : Gerente em ambas as colunas, procura colocar dessa forma: Monitoramento e Controle das atividades.... e na outra coluna: Monitoramento dos projetos e da operação continuada...

Comment [T558]: A tabela está quebrada. Faz ela de forma que não fique assim. E se passar de uma página para outra, que a linha com o cabeçalho dela apareça também nessa página. Ela deve ser centralizada.

A seguir, o foco principal de um Programa é a obtenção de benefícios que tendem a estar totalmente aderentes aos objetivos estratégicos e corporativos (benefícios como resultados de estratégia), onde a Gestão de Programas deve abranger 3(três) dimensões que são: [PMI 2006]

Comment [T559]: Sempre que começa alguma seção, não tem esse espaçamento inicial. Verifica isso ao longo do texto..

Comment [T560]: Coloca só por extenso, tira os parênteses e o número 3.

Comment [T561]: Não seria bom citá-los aqui já, para o leitor já tem uma idéia inicial do que virá nas próximas subseções?

Considerado o principal foco (do ponto de vista de resultados) da Gestão de Programas, consiste em definir um plano formalizado na qual a Organização estabelece quais os benefícios são esperados como resultados da implementação da Gestão de Programas. Neste plano, além de aspectos como a própria visão dos benefícios e resultados esperados, também devem ser estabelecidas métricas e procedimentos para acompanhamento das metas, definição de regras e responsabilidades, planos de comunicações e de transições das fases de Programas e ou Projetos para os ambientes de operações (ongoing).

Comment [T562]: Acho que não precisa desse parênteses.

Ações relativas: Avaliar o valor e o impacto do programa na organização; analisar os impactos das mudanças; identificar as interdependências dos benefícios intermediários que serão entregues pelos projetos que compõem o programa. Garantir que os benefícios sejam:

Comment [PRCA563]: Acharia mais interessante que fizesse uma pequena introdução às ações relativas, do que já jogá-las no texto de bate-pronto

Possuir um bom modelo de gestão capaz de envolver, comunicar e até mesmo influenciar os *stakeholders*, visando atingir as metas do Programa é de suma importância para alcançar o sucesso do Programa. Lembrando, mais uma vez, que a principal dimensão a ser atendida na Gestão focada em Programas são os Benefícios a serem obtidos pela organização, onde diversos *stakeholders* podem e devem ter suas atenções voltados ao Programa. Portanto, gerenciar de forma efetiva é de vital importância para o sucesso do Programa como um todo.

Comment [T564]: É infinitivo mesmo, ou colocasse um R a mais? 😊

Comment [T565]: Está faltando algo pra ligar a frase anterior com essa, não?

Comment [T566]: Não tem esse a aqui não.

Comment [T567]: A concordância deveria ser: voltadas

- Aproveitarão dos benefícios;

Comment [T568]: Esses itens de numeração devem estar padronizados. Ora você usa bolinha, ora tracinho. Acredito que não possa ser dessa forma.

Comment [T569]: Os benefícios.

Neste assunto dos *stakeholders* existem interesses ou objetivos conflitantes, e cabe a equipe de gerenciamento do programa administrar essas expectativas. Principais *stakeholders*:

Comment [T570]: Que expectativas? Se estás falando dos interesses e objetivos conflitantes, aho que essa palavra não ficou legal aí. Troca por outra: sei lá, pode ser: esses conflitos.

- **PMO.**

Comment [T571]: É bom colocar por extenso também o que significa PMO e em itálico o significado.

Com um modelo de Governança estabelecido, as equipes e envolvidos no Programa se utilizarão deste **framework** como guia para a gestão, visando uma grande probabilidade de sucesso em busca dos objetivos traçados pela Organização, permitindo executar, monitorar e controlar todas as fases de um Programa, acompanhando os investimentos, progressos, desvios, ações e principalmente os benefícios, quando estes começarem a se evidenciar.

Comment [T572]: Deve estar em itálico.

Comment [T573]: Isso é uma função do Comitê?

- Provê orientação e toma decisões relativas à mudanças que afetarão os resultados

Comment [T574]: Deveria estar no plural: às

- As revisões realizadas são maneiras mais eficiente para avaliar a performance

Comment [T575]: Função do Comitê? Tenta mudar a forma de escrever, para ficar parecendo como função mesmo. Por exemplo: Avaliar o desempenho de maneira mais eficiente através das revisões realizadas.

Comment [T576]: Essa palavra acho que não existe em Português. Deve ser desempenho.

Para cada fim de fase existirá Portões de revisão, onde o Comitê revisará e depois informará se pode ou não passar para a próxima fase. Os Portões de revisões serve para:

Comment [T577]: Concordância deve ser: servem

- Set up *Pré-programa*;

Comment [T578]: Em itálico

Programs						
Pre-Program Set Up	Program Set Up	Establish Program Management & Technical Infrastructure	Deliver Incremental Benefits	Close the Program	Transition	Ongoing Operations

Comment [T580]: Deve estar centralizada. As palavras da figura não estão muito boas de serem lidas. Acho que deverias fazer outra e colocar mais legível a escrita.

- **Fase 1: *Set up* Pré-Programa**

Nesta fase acontece a aprovação para o programa; entrega dos benefícios esperados; alinhamento dos objetivos estratégicos da organização; Resposta do gerente do programa para o *por que* dos benefícios esperados alcançados através da criação de um programa em vez da implantação de um projeto.

Comment [T581]: Em itálico

Comment [T582]: Deve ser junto e com acento.

Comment [T583]: Fontes misturadas no texto. Padronizar.

- **Program Charter**, que documenta a visão, objetivos, benefícios, restrições e premissas;

Comment [T584]: Em itálico

- **Fase 2: Set up Programa**

Comment [T585]: Em itálico

- **Riscos:**

Comment [T586]: Resultados Esperados?

Inicia-se os projetos do programa; coordenar seus produtos para criar benefícios incrementais; maior parte do trabalho do programa através de seus componentes é iniciada; é iterativa; pode ter duração ilimitada; atividades são repetidas tantas vezes quantas necessárias e benefícios são atingidos cumulativamente. Principais atividades:

Comment [T587]: Concordância correta seria: Iniciam-se

Comment [T588]: Esse parágrafo deveria ser reescrito, porque está muito topificado.

- Benefícios esperados; |
- Garantir que as atividades comuns e dependências entre os projetos e/ou

Comment [T589]: Atividades.. coloca um verbo que deve ser feito com essas duas coisas.

Executa um fechamento controlado do programa; desativação da organização do programa e sua infra-estrutura; transição para outros grupos. Principais atividades:

Comment [T590]: Escreve em um texto corrido, sem esses itens citados entre ;

São muito similares aos processos de um projeto. Porém, tratam questões a um nível mais alto. Envolvem menos detalhes; resolvem questões entre projetos; entregamos benefícios do programa. Componentes comuns no processo do gerenciamento de programa:

Comment [T591]: O texto deve ser impessoal. Não pode usar primeira pessoa.

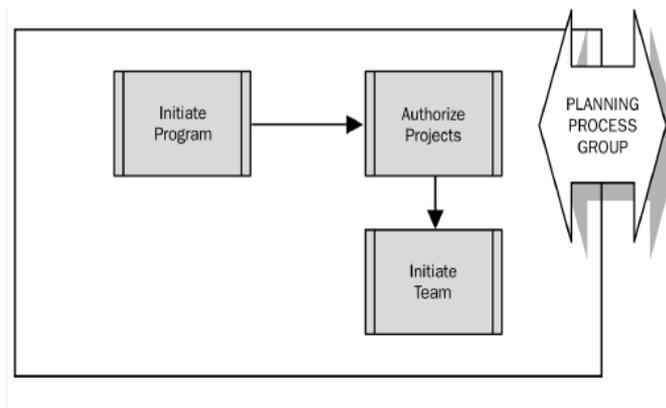
ENTRADA
<ul style="list-style-type: none">• Declaração de escopo do programa• Critério de seleção dos projetos• Plano estratégico

SAÍDA
<ul style="list-style-type: none">• Requisitos de relatórios do Programa• <i>Project Charter</i>• Designação do gerente do projeto• Identificação do Patrocinador do projeto• Aprovação das reservas financeiras do projeto

- Responsável por designar e alocar os recursos humanos

ENTRADA
<ul style="list-style-type: none"> • Prática de recrutamento • Descrição de recursos

SAÍDA
<ul style="list-style-type: none"> • Núcleo do time do programa designado • Gerente do programa designado • Time do Programa



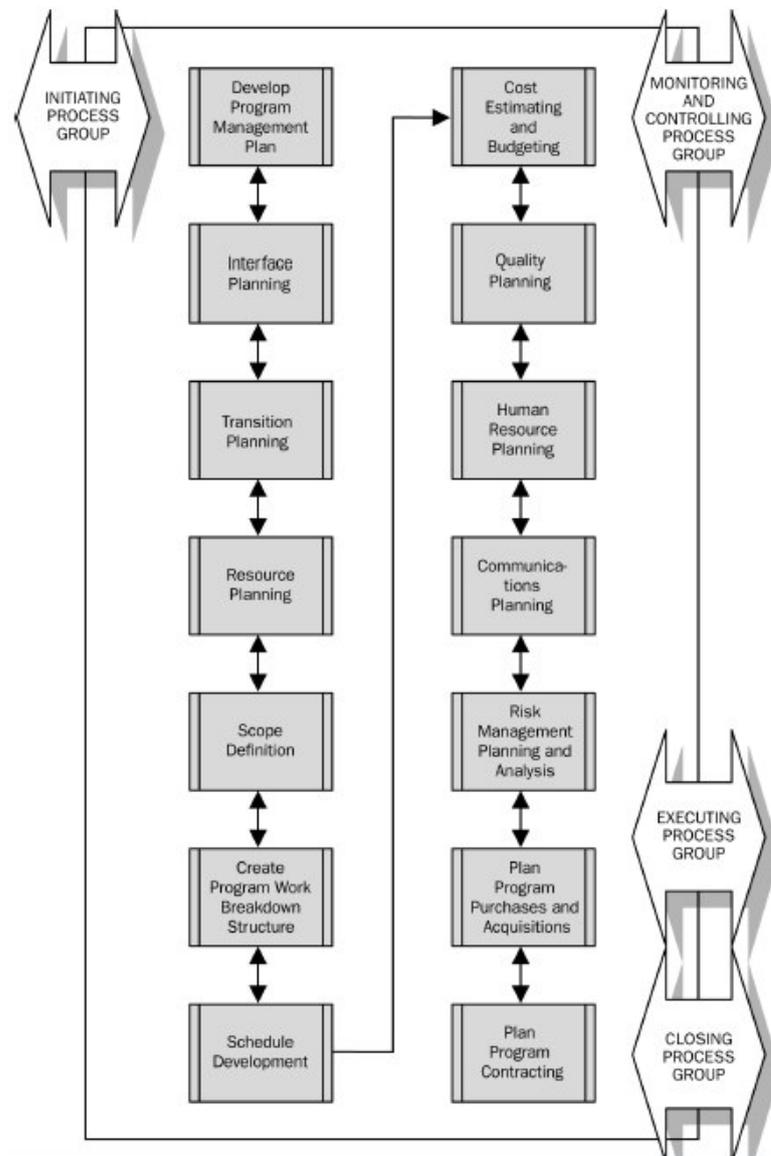
Comment [T592]: Figura deve estar em Português. Centralizada. E contextualizada no texto.

Aqui, a atenção está voltada em construir a base do programa visando prepará-lo para uma execução eficiente. O grupo é responsável por formalizar o escopo do trabalho a ser desenvolvido; identificar as entregas que irão satisfazer os objetivos do programa; entregar os benefícios esperados; orçamento; comunicação; **escopo, custos, tempo**, riscos associados; planos para definir as métricas e o procedimento para monitorar a realização e o alcance dos benefícios.

Comment [T593]: Usa ou ; ou , mas não pode usar os dois misturados.

5) **Definição do Escopo** - Desenvolver uma declaração de escopo detalhada do

Comment [T594]: Por que está destacado?



Comment [T595]: A Figura deve estar em Português.

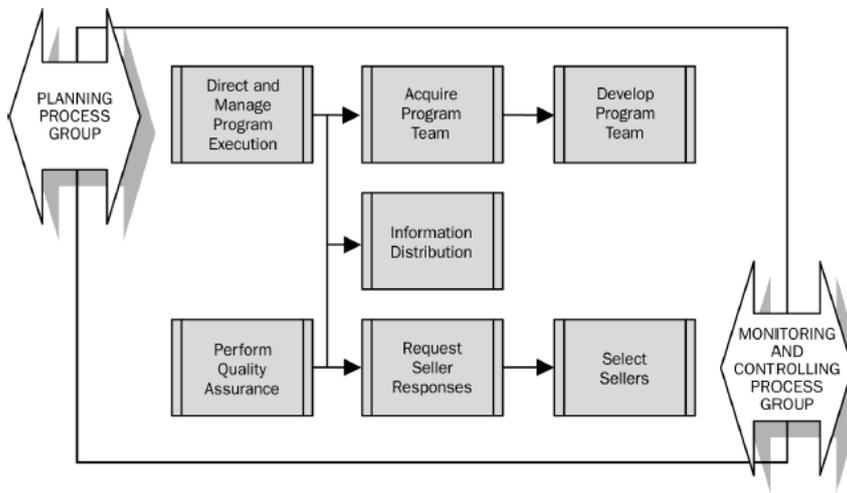
Direcionam a realização do esforço do programa; gerenciam custos, prazos e qualidade; provêm informações sobre o andamento do programa; solicitações de mudança para o grupo de processos de monitoramento e controle; garantem que os *stakeholders* estejam recebendo as informações solicitadas na periodicidade definida previamente.

Comment [T596]: Esse acento não existe mais

Comment [T597]: Texto com outra fonte. Deve estar justificado. Times New Roman 12.

- 7) **Selecionar Fornecedores** - Rever as ofertas (valores), negociar os detalhes do contrato, incluindo termos técnicos e condições, papéis, responsabilidades, entregas e preço final.

Comment [T598]: Por que ele está destacado?



Comment [T599]: Figura deve estar em Português.

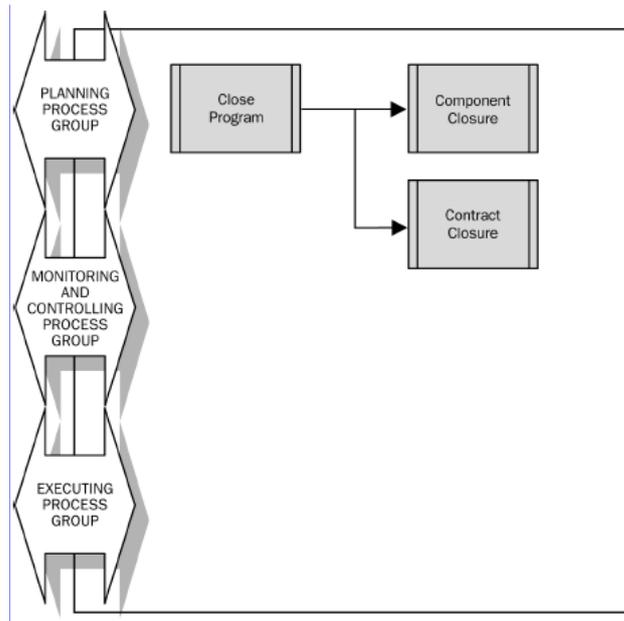
Os processos do grupo de execução:

Comment [T600]: Não seria Monitoramento e Controle?

- 11) **Controle e Monitoramento de Riscos** - Rastrear os riscos identificados e identificar novos, executar o plano de resposta aos riscos e avaliar sua efetividade em reduzi-los através do ciclo de vida do programa.

Comment [T601]: Por que apenas um destacado?

- 3) **Encerramento dos Contratos** - Verificação do produto e atualização de todos os registros contratuais.



Comment [T603]: Fiugura deve estar em Português, centralizada.

- **Gestão de Portfólios associado à Gestão de Programas**

Comment [T604]: Ainda não está no formato obrigatório exigido neste tópico.

- **Modelo PMI - 2006**

Comment [T605]: Mesmo comentário de Tópicos de Pesquisa.

- **Referências**

[ALENCAR 2005] **Análise de Risco em Gerência de Projetos**. 1ª Edição, 2005, Antonio Juarez Alencar / Eber Assis Schmitz, Editora Brasport

Comment [PRCA606]: As Referências não possuem número de seção (ver template)

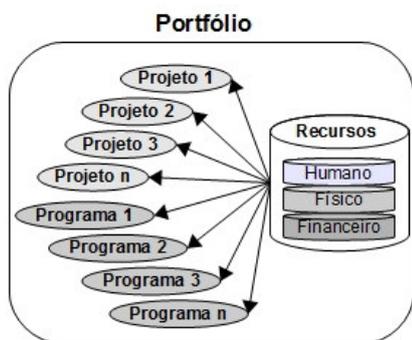
Comment [T607]: Referências fora do padrão. Verificar padrão SBC. Poucas referências ao longo de todo o texto..

Comment [PRCA608]: Dispor as referências como no template

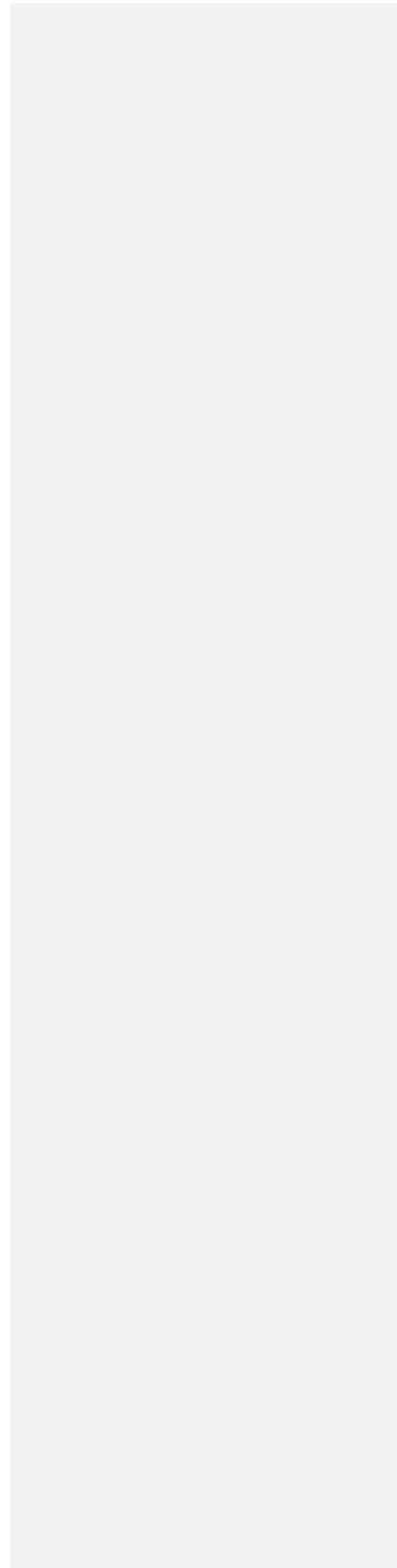
Comment [T609]: Análise

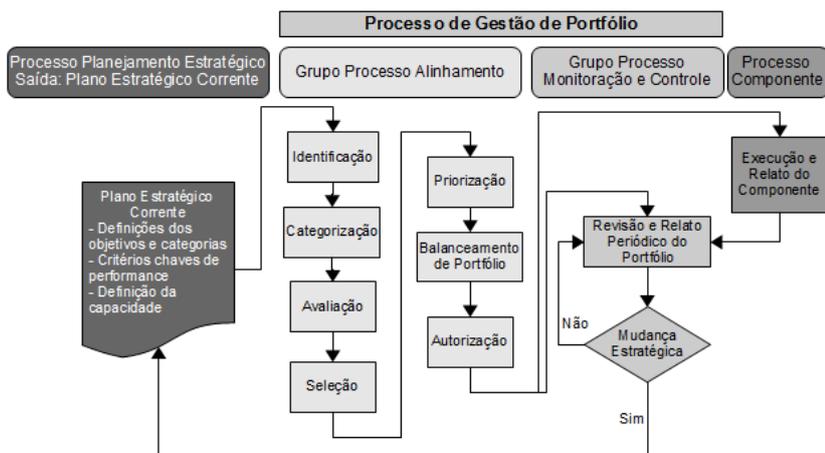
Sumário

15.13 REFERÊNCIAS BIBLIOGRÁFICAS.....	120
---------------------------------------	-----



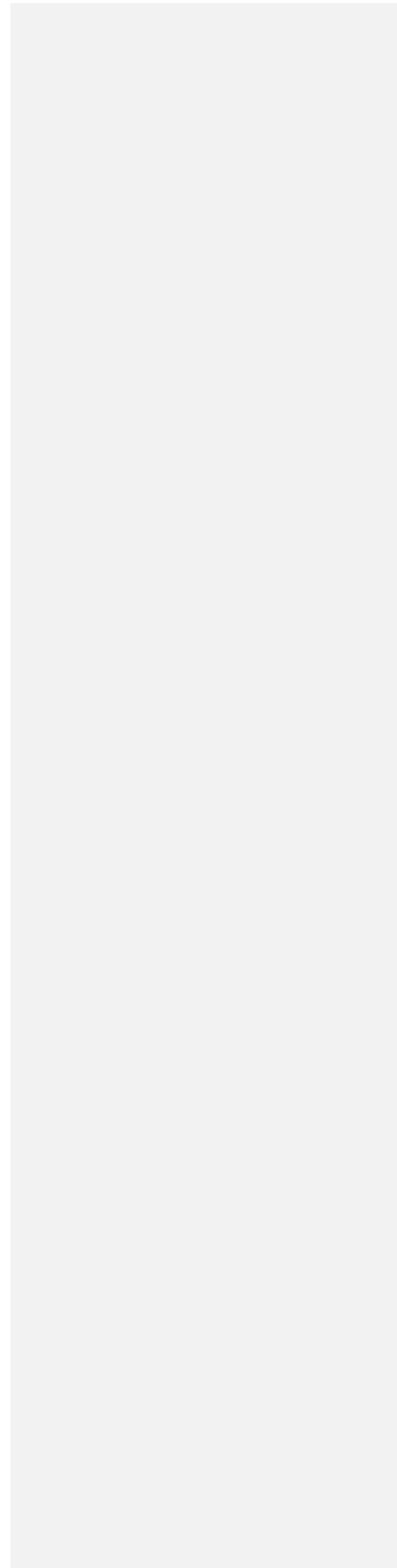
Primeiramente, é necessário **DEFINIR CLARAMENTE QUAIS SÃO** os objetivos estratégicos organizacionais. Atualmente existem diversas ferramentas que auxiliam no desenvolvimento da estratégia organizacional. Uma delas é a Análise das Forças, Fraquezas, Oportunidades **[REMOVER ESPAÇO]** e Ameaças, também conhecida como Análise SWOT. Essa ferramenta auxilia na definição dos objetivos que a organização deverá atingir para maximizar suas forças, minimizar suas fraquezas, aproveitar as oportunidades e tratar as

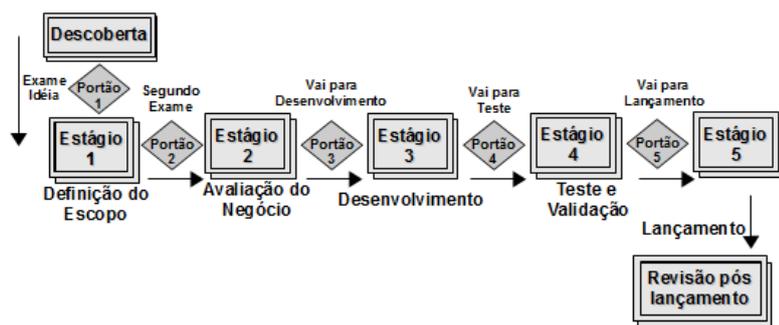


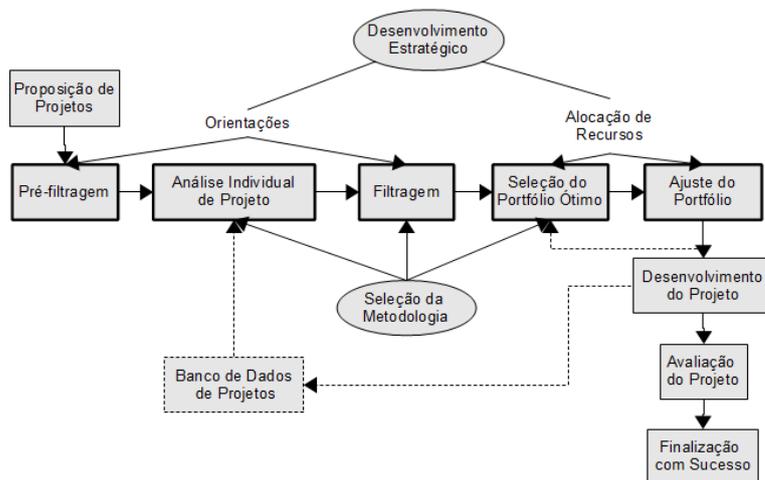


O objetivo deste processo é separar os componentes em categorias empresariais relevantes, de modo que um conjunto comum de filtros e critérios de decisão possam ser aplicados durante[**s REMOVE**R] os processos de

avaliação, seleção, priorização e balanceamento. Essas categorias são definidas com base no plano estratégico. Os componentes de uma determinada categoria terão um objetivo comum e poderão ser medidos com base nesse objetivo, independentemente da sua origem na organização. A categorização dos componentes permite que a organização eventualmente possa equilibrar os seus investimentos e seus riscos entre todas as categorias e objetivos estratégicos.





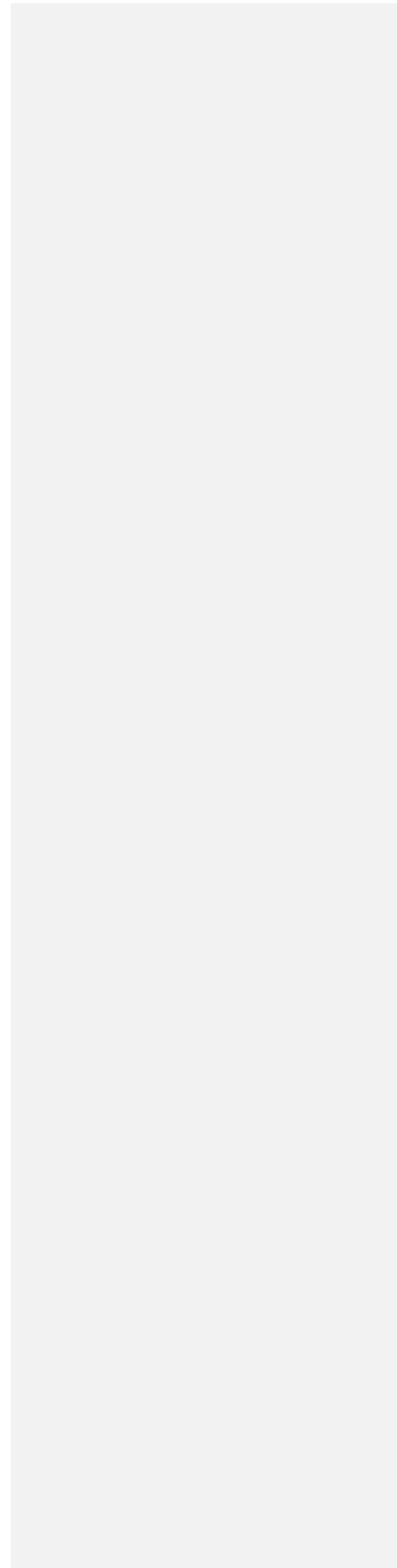


Nesta fase, **que ocorre após a de análise individual dos projetos, são verificados criteriosamente os parâmetros produzidos na análise. Essa filtragem servirá de base para a seleção dos** projetos, assim como, para eliminar

quaisquer projetos que não preencham critérios pré-estabelecidos. Vale ressaltar que projetos obrigatórios ou necessários para apoiar outros projetos não serão eliminados.

Através DO sistema citado no tópico anterior, também são realizadas os ciclos de pareceres. TAIS CICLOS são atividades executadas periodicamente com o objetivo de realizar o acompanhamento dos projetos. Durante ESTAS ATIVIDADES, cada gerente de projeto emitE um parecer sobre seu projeto. Neste parecer estão informações sobre

potenciais problemas do projeto, tais como: escassez de recurso, falta de apoio, conflitos ou qualquer outro fato que esteja ou possa comprometer o andamento das atividades.



[SERPRO 2009] <http://www.serpro.gov.br>

Este capítulo descreve uma estrutura que surgiu a partir do reconhecimento da importância de uma gestão de projetos mais eficiente nas organizações. Tal importância é atribuída principalmente ao efeito positivo que as atividades que envolvem gestão de projetos têm dado sobre a lucratividade nos negócios. Diante disto, estas atividades acabaram demandando uma maior ênfase no profissionalismo o que culminou na necessidade de criação de departamentos especializados. Tais departamentos foram denominados Escritório de Projetos (PO – *Project Office*) ou Escritório de Gestão de Projetos (PMO – *Project Management Office*), e serão apresentados neste capítulo. Aqui serão descritos seus principais papéis, funções, objetivos, classificações, boas práticas de sua implantação e um estudo de caso demonstrando sua relevância, no que diz respeito a gestão de projetos, nas organizações contemporâneas.

Comment [Carlos AI610]: organizacional

Comment [Carlos AI611]: corporações

Comment [Carlos AI612]: causado

Comment [Carlos AI613]: disso

O Escritório de Projetos ou *Project Management Office* (PMO) tem sido uma das estruturas de Gerenciamento de Projetos mais exploradas atualmente pelas organizações. Isto tem se dado principalmente pela necessidade destas organizações possuírem uma entidade de apoio interna que preste suporte as atividades de implementação de princípios, práticas, metodologias, ferramentas e técnicas em Gerenciamento de Projetos [Marrom 2009].

Comment [Carlos AI614]: isso

Brian Hobbs e Monique Aubry [Hobbs e Aubry 2007] realizaram uma pesquisa pela internet em três etapas entre 2007 e 2008, envolvendo 500 organizações do Canadá, EUA e Europa. Tal pesquisa tinha como principal objetivo melhor entender o PMO e o valor agregado em sua implantação nas estruturas organizacionais. Dentre os principais destaques apresentados durante pesquisa foram listados:

Comment [Carlos AI615]: Colocar a referência no final do parágrafo

Comment [Carlos AI616]: Vírgula

O mercado contemporâneo tem forçado as organizações a definir estratégias gerenciais cada vez mais arrojadas na busca por uma maior produtividade dos projetos, maior eficiência dos processos e uma consolidação efetiva de informações estratégicas a serem repassadas a alta administração. Entre outros papéis e funções atribuídas atualmente ao PMO, podemos dizer que prover suporte a estes problemas é apenas um deles.

Comment [Carlos Al617]: pode-se

O papel dos PMOs tem se mostrado diverso e variado em nossa atual realidade. Ajuste de padrões e de metodologias para as gerências de projeto tem sido um dos grandes focos desta estrutura. Atividades ligadas a gerência de recursos humanos e também a responsabilidade na execução dos projetos também tem sido fortemente atribuídas ao PMO [Ribero 2007]. Além disto, as funções de unificar as estratégias e disseminá-las na organização como um Gerenciamento de Projetos responsável por preencher a lacuna entre a visão da empresa e de seus projetos, tem também sido atribuído a este tipo de estrutura [Kerzner 2003].

Comment [Carlos Al618]: disso

Comment [Carlos Al619]: Ficou estranho

Diante do que vem sendo apresentado neste capítulo nos resta um questionamento: Enfim, qual seria o principal objetivo dos PMOs? Os objetivos básicos, segundo [Dinsmore 2003] são: “orientar e dar suporte aos gerentes de projetos permitindo à empresa desenvolver seus projetos da forma mais eficiente e eficaz possível”.

Comment [Carlos Al620]: ...fica o seguinte quest..

- Tornar possível a execução dos projetos de forma que os mesmos fiquem alinhados com os objetivos da alta direção. Devido a disposição do PMO, torna-se possível a concentração das informações e condução dos projetos, facilitando assim o alinhamento entre os objetivos dos projetos e os objetivos organizacionais.

Comment [Carlos Al621]: vírgula

Podemos observar então que os objetivos de um PMO ainda são os mais variados e ainda não estão padronizados nas organizações. Todos os objetivos apresentados nesta seção buscam na realidade trazer uma melhoria na eficiência do planejamento e na condução dos projetos. Neste sentido, uma seleção e análise sobre os projetos existentes de forma ágil, torna-se de vital importância as organizações. Entregar os projetos no prazo, dentro do orçamento e atendendo aos objetivos de negócio da organização são critérios de suma importância segundo [Dai 2003].

Comment [Carlos Al622]: observa-se

- Escritório de Projetos de Grupo de Clientes;
- E Escritório de Projetos Corporativo.

Comment [Carlos Al623]: Tirar o "e"

O primeiro é caracterizado por gerenciar um conjunto crucial de recursos, dentro de um determinado setor, para a utilização em projetos peculiares do setor na organização. Os Escritórios de Projetos de Grupo de Clientes por sua vez, tem como objetivo principal progredir no diálogo com os clientes. Projetos de propósitos iguais e clientes são reunidos e acabam funcionando como uma organização provisória dentro da própria organização. Desta forma, segundo esta estrutura, pode existir vários escritórios de projeto de grupos de clientes sendo executados ao mesmo tempo na organização. Por fim, os Escritórios de Projetos Corporativos apresentam como principal característica acolher toda a organização concentrando-se nas questões estratégicas e corporativas da mesma. Neste sentido, este tipo de escritório tenta fazer com que todas as questões proeminentes para o sucesso dos projetos sejam discutidas e analisadas em seu âmbito, para que seja possível prestar uma maior assistência aos gerentes sobre suas decisões.

Comment [Carlos Al624]: Nesse

Comment [Carlos Al625]: Sigla já explicada

resultados é da equipe de projeto que está no comando do mesmo, sejam esses resultados positivos ou negativos. Este desenho ocorre caracteristicamente em circunstâncias em que o projeto não tem um relacionamento estreito com as outras partes da organização e a empresa não tem muita experiência em Gerenciamento de Projetos.

Vargas [Vargas 2003] afirma que este tipo de escritório de projetos destina-se ao Gerenciamento de um projeto ou programa muito específico. Neste cenário, as práticas de gerenciamento de projetos são provenientes da experiência do líder do projeto, não havendo um apoio nem mesmo cultura pela organização. A estrutura da APT tem a responsabilidade de gerenciar todas as áreas dos projetos, de forma centralizada.

Segundo [Dinsmore 2003], existem pontos cruciais para o sucesso de uma Equipe de Projeto Autônoma:

Comment [Carlos Al626]: Letra minúscula

Comment [Carlos Al627]: Tirar a vírgula

Figura 1 - APT: Equipe Autônoma de Projeto – adaptado por Gerhard (Gerhard 2004) de Dinsmore (Dinsmore 2003) e Vargas (Vargas 2003).

No formato Escritório de Suporte de Projetos ou *Project Support Office*, é apresentado um apoio técnico ao projeto. Segundo este tipo de estrutura, [Vargas 2003] enfatiza que um PSO, destina-se ao suporte de diversos projetos concomitantemente, dando suporte especializado através do uso de ferramentas e recursos. De acordo com Dinsmore [Dinsmore 2003] e Vargas [Vargas 2003], um PSO em sua essência pode apresentar as seguintes opções de serviços:

Comment [Carlos Al628]: A resolução da figura não está legível.

Comment [Carlos Al629]: Verificar a numeração da figura

Comment [Carlos Al630]: Substituir pela sigla PSO

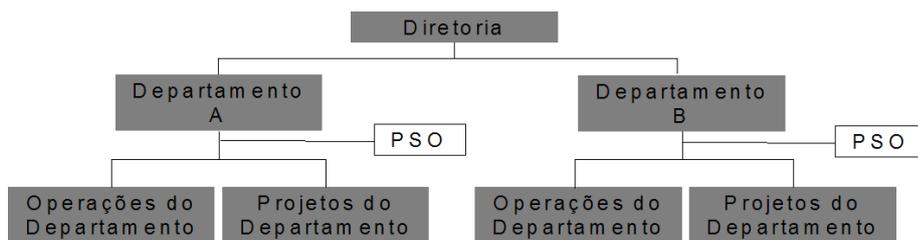


Figura 2 – PSO por departamento segundo Vargas [Vargas 2003].

Comment [Carlos Al631]: Figura ilegível

No modelo seguinte é apresentado o escritório de *Enterprise Project Support Office*. Segundo o autor neste cenário o escritório de projetos assume uma posição de planejador estratégico dos projetos da organização estando diretamente ligado ao alto escalão da organização.

Comment [Carlos Al632]: Colocar em português seguido da sigla em inglês

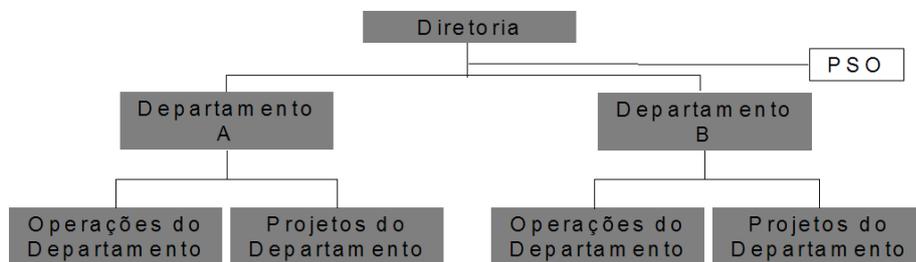


Figura 3 – PSO Corporativo segundo Vargas [Vargas 2003].

Comment [Carlos Al633]: Figura ilegível

O Centro de Excelência em Gestão de Projetos ou (*Project Management Center Of Excellence*), apresenta como principal característica a fundamental referência em termos de conhecimento em gestão de projetos para a organização. Nada mais justo e necessário a um centro de excelência em Gestão de Projetos. Todavia não são de sua responsabilidade os resultados apresentados pelos projetos. O PMCOE é a estrutura responsável por disseminar a cultura de Gerenciamento de Projetos e manter as metodologias aplicáveis ao âmbito organizacional.

Comment [Carlos Al634]: Substituir pela sigla

Comment [Carlos Al635]: Tirar o “)”

O Escritório de Gerência de Programas ou *Program Management Office* por sua vez apresenta como característica a gerência dos Gerentes de Projetos. Sendo assim, a responsabilidade sobre os resultados dos projetos recai sobre sua competência. Outro nome comumente atribuído a este tipo de estrutura é o de gerência de portfólio de projetos. O PrgMO tenta unir as funções do PMCOE e do PSO. Neste caso, projetos geridos isoladamente por uma determinada divisão, têm apoio do PrgMO sempre que preciso. Para que ele funcione com perfeição é preciso que tenha: poder, precedência corporativa e autoridade em âmbito empresarial.

Comment [Carlos Al636]: Substituir pela sigla

Figura 3 – PrgMO Corporativo segundo Vargas (2003).

Comment [Carlos Al637]: Figura ilegível

Para Block é primordial se identificar o patrocinador antes de tudo. Este é peça chave

funções e montagem do plano de comunicação a ser utilizado pelo PMO. Em seguida pode-se preparar o orçamento e tentar levantar os fundos necessários, inclusive a parte que cabe ao patrocinador, iniciando então logo em seguida o projeto piloto para avaliá-lo durante operação.

De acordo com Vargas [Vargas 2003], o primeiro passo é escolher o tipo de PMO que deve ser implantado na organização. Só aí então se pode levantar um patrocinador que lhe suporte. O próximo passo é criar uma estrutura com todas as possíveis dependências para que o PMO funcione. Motivar e promover o repasse de conhecimento entre os envolvidos no projeto é o passo posterior acompanhado de uma melhoria contínua em toda infra-estrutura. Após todos estes procedimentos pode-se de fato fundar um projeto piloto colocando-o em operação sempre o monitorando e observando quais os possíveis pontos de melhoria contínua.

Comment [Carlos Al638]: Frase estranha, reformular.

- **Repositório:** o PMO deve estruturar e gerenciar os documentos do repositório, para que seja possível reutilizar processos, procedimentos, *templates*, etc. [20].

Comment [Carlos Al639]: Tirar espaço

- Garantir que todo o trabalho realizado seja autorizado e tenha respaldo por documentos contratuais.

Comment [Carlos AI640]: Tirar espaço

O mercado de finanças públicas é ramo de atuação do SERPRO, o qual é composto atualmente pelo Ministério da Fazenda com suas secretarias e demais órgãos. Outro segmento igualmente importante é o de ações estruturadoras e integradoras da Administração Pública Federal cuja gestão e articulação compete ao Ministério do Planejamento, Orçamento e Gestão [Porto e Melo 2008]. A empresa atualmente destaca-se pelos seus pouco mais de 40 anos de atuação no mercado e por ser uma organização de destaque do governo federal para o segmento de tecnologia de

informação possuindo algo em torno de 10 mil funcionários e receita anual da ordem de R\$1,5 bilhões-ano [Magalhães e Gusmão 2009].

Entre os principais clientes do SERPRO podemos citar: Presidência da República, Advocacia-Geral da União, Casa Civil da Presidência da República, Controladoria Geral da União, Ministério da Fazenda, Ministério da Educação, Ministério do Planejamento, Orçamento e Gestão, Ministério do Desenvolvimento, Indústria e Comércio Exterior, Ministério do Trabalho e Emprego, Ministério da Integração Nacional, Ministério das Minas e Energia, Ministério da Defesa, Ministério dos Transportes, Ministério da Saúde, Ministério das Cidades, Ministério da Justiça, Ministério do Desenvolvimento Agrário, Ministério dos Esportes.

Comment [Carlos Al641]: estão:

[Magalhães e Gusmão 2009] acrescentam ainda que a empresa em 2003, através de decisões de sua própria diretoria, saiu do Orçamento Geral da União (OGU), visando maior independência frente à questão comercial e financeira. O papel do OGU até então

dependeria exclusivamente de sua eficiência com a possibilidade de geração de receita própria.

suporte dos Gestores de Projetos e controle do desempenho dos projetos. Também foi criado o papel de Gestor de Projeto com a função de gerenciar e conduzir um projeto, o que inclui, liderança, planejamento e gestão do desenvolvimento de todos os resultados entregáveis do projeto. Com o objetivo de identificar as ações necessárias, foi considerado o contexto do gerenciamento de projetos adotados em nível corporativo, onde várias questões foram observadas de forma integrada e o encadeamento das ações foi uma premissa para garantir o sucesso da iniciativa.

Diante disto, decidiu-se decompor o processo em fases administráveis, baseadas em

Comment [Carlos AI642]: Desativar a

adequadas ao SERPRO, com entrega de produtos bem definidos e com o envolvimento de todas as unidades desde o início, em trabalhos conduzidos por equipes

1.6.3.2 Fases

As fases foram organizadas considerando, também, os três pilares fundamentais em qualquer estratégia: processos, pessoas e tecnologia. Neste sentido, a primeira fase estava focada no amadurecimento do conhecimento e criação dos fundamentos básicos de Gerenciamento de Projetos na Empresa, o que se traduziu na elaboração de uma Política Corporativa, de um Processo de Gerenciamento de Projetos e de um Programa Corporativo de Implantação do gerenciamento de projetos no Serpro. Realização de benchmarking, disseminação (palestras, aquisição de livros especializados no assunto pelo CDI e criação da comunidade de Gestão de Projetos no SERPRO.) e conhecimento do uso atual de práticas de gerenciamento de projetos nas unidades do SERPRO ocorreram nesta fase inicial, em paralelo à capacitação em Gerência de Projetos, baseada nas práticas do PMI – *Project Management Institute*, com o objetivo de nivelar o grau de conhecimento e conceitos dos representantes das Unidades.

A segunda fase constituiu-se da instituição da política e do processo corporativo de gerenciamento de projetos e focava em ações de conscientização e realização de projetos pilotos, inclusive para criação do Portfólio de Projetos. Em paralelo, ações de planejamento, de contratação de treinamentos e de especificação de ferramenta para apoio ao gerenciamento de projetos foram planejadas. Foi uma fase de curta duração, cujos resultados foram úteis para capacitação e iniciativas de implantação de gerenciamento de projetos nas Unidades na fase posterior. Foram previstas ações para divulgação maciça, instituição do Comitê Gestor, com representantes das Unidades e do

Escritórios de Suporte e Controle de Projetos, aplicação do processo corporativo em pequenas iniciativas das Unidades envolvendo os profissionais que foram treinados e criação de um guia para orientar o planejamento de implantação pelas Unidades na próxima fase.

Na terceira fase, o foco estava na implantação efetiva do processo nas unidades, sob coordenação do Escritório de Suporte e Controle de Projetos da Unidade e acompanhamento pelo Escritório Estratégico de Projetos. Um Plano de ações foi elaborado por cada Superintendência, com base no guia elaborado na segunda fase, prevendo treinamentos seguidos de aplicação em projetos reais não críticos e de curto prazo, servindo como marco efetivo da implantação da cultura na Unidade. Foi uma fase com maior tempo de duração, pois previa também o desenvolvimento ou a aquisição de ferramenta para apoio ao gerenciamento de projetos, que dependia de processo de licitação, promoção de oficinas de trabalho para aplicação de técnicas adotadas no processo corporativo e incentivo a certificação PMP – *Project Management Professional*.

Na quarta fase, o Escritório de Suporte e Controle de Projetos das Unidades, assessorado pelo Escritório Estratégico de projetos, deveria dar ênfase à extensão da implantação do processo corporativo a projetos de toda natureza realizados pela unidade, ao uso das lições aprendidas e à certificação PMP de seus profissionais. Todos os projetos deveriam ser contemplados no Portfólio, o acultramento em gerenciamento de projetos deveria

Comment [Carlos AI643]: Desativar a hifenização

Comment [Carlos AI644]: Desativar a hifenização

Comment [Carlos AI645]: Desativar a hifenização

Comment [Carlos AI646]: Desativar a hifenização

objeto de avaliações do processo corporativo em busca de melhoria e da viabilidade de integração com as demais ferramentas existentes no Serpro.

Na quinta fase, o foco era na transição para a melhoria contínua, onde as ações deveriam ser direcionadas para a melhoria dos processos, atualização de software e meios de comunicação, reciclagem da equipe e avaliação da maturidade da Empresa em

Comment [Carlos AI647]: Desativar a hifenização

- Melhoria na qualidade da gestão dos projetos conduzidos pela organização, com impacto positivo percebido, inclusive, pelo cliente, que começa a requisitar proativamente os artefatos e atividades previstas no Processo de Gerenciamento de P

Comment [Carlos AI648]: Desativar a hifenização

Comment [Carlos AI649]: Tirar o enter

1.6.5 Melhoria Contínua

Durante o primeiro semestre de 2008, enquanto encerrava-se a 4ª fase, realizou-se a transição do projeto de implantação para o momento de melhoria contínua. A definição da estratégia de evolução deu-se essencialmente através de discussões entre Escritório Estratégico de Projetos e a alta liderança da empresa. Como produto das discussões foi criado o Sistema de Gerenciamento de Projetos do Serpro (SGPS) compreendendo todo o conjunto de componentes responsáveis pela execução e evolução da solução de gerenciamento de projetos da empresa, como, por exemplo, estruturas funcionais, cursos, processos, ferramentas, etc. Também foram criados o Processo de Estruturação do SGPS, que agrega todas as ações de evolução do SGPS, e o Grupo de Trabalho de Estruturação do SGPS (GTSGPS), grupo inter-área de funcionários que assumiu a responsabilidade por conduzir as atividades do processo. Tanto o processo quanto o GT-SGPS são coordenados pelo Escritório Estratégico de Projetos.

O Processo define um canal para controlar o ciclo de vida das demandas de melhorias através do sistema corporativo para gestão de demandas da empresa (Sistema

Comment [Carlos AI650]: Desativar a hifenização

Comment [Carlos AI651]: Desativar a hifenização

Continuamente a coordenação do grupo monitora mudanças no Balde, na situação do

cenário empresarial. A título de ilustração, descrevemos as principais demandas selecionadas para implantação em 2008:

Comment [Carlos Al652]: descreve-se

Durante a escrita deste capítulo foram identificadas algumas linhas de pesquisa desenvolvidas pela comunidade de Gerência de Projetos no que diz respeito a escritório de projetos. Entre os principais tópicos apresentados podemos citar:

Comment [Carlos Al653]: estão:

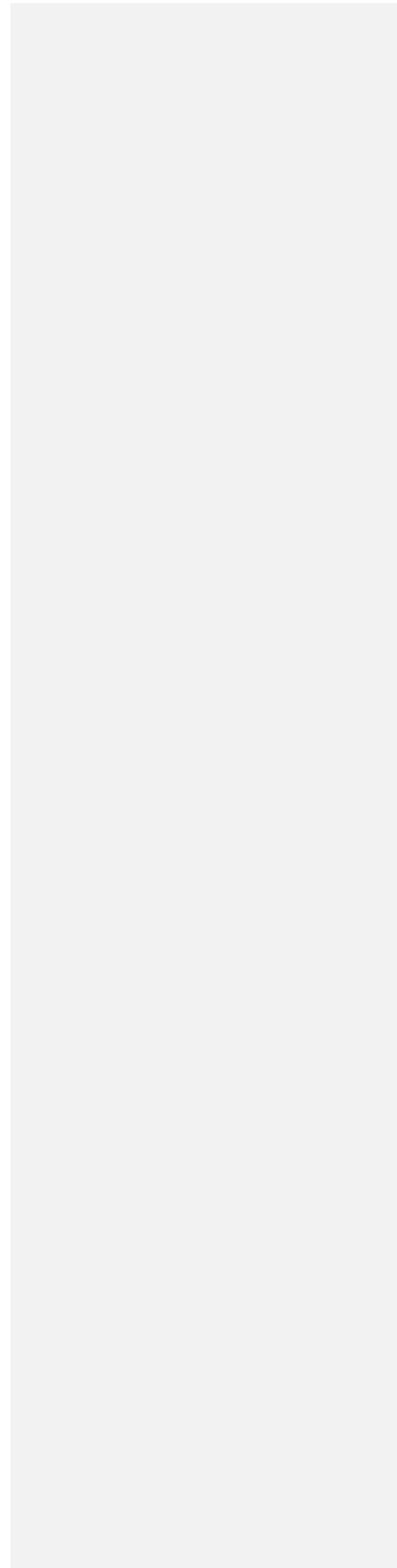
- Analisando a Figura abaixo, a qual representa um organograma organizacional de uma empresa fictícia, que tipo Escritório de Projetos melhor se enquadraria segundo a caixa destacada em cinza?

Comment [Carlos AI654]: Figura ilegível

- Analisando a Figura abaixo, a qual representa um outro organograma organizacional de uma segunda empresa fictícia, que tipo Escritório de Projetos melhor se enquadraria segundo a caixa destacada em cinza?

Comment [Carlos AI655]: Figura ilegível

[Magalhães e Gusmão 2009] MAGALHÃES, Edviges M. C. & GUSMÃO, Cristine M. G. Identificação dos principais fatores de riscos na implantação de um PMO em uma organização com foco em tecnologia da informação. Monografia - Programa de Pós- Graduação em Engenharia da Universidade de Pernambuco – Especialização em Gestão Global de Projetos. 2009.



<u>20.1. INTRODUÇÃO A MATURIDADE EM GESTÃO DE PROJETOS</u>	<u>102</u>
<u>20.2. MODELOS DE MATURIDADE EM GESTÃO DE PROJETOS.....</u>	<u>103</u>
20.2.1. ORGANIZATIONAL PROJECT MANAGEMENT MATURITY MODEL - PMI.....	103
20.2.2. PROJECT MANAGEMENT MATURITY MODEL – PM SOLUTIONS.....	104
20.2.3. MODELO DE MATURIDADE EM GERENCIAMENTO DE PROJETOS – DARCI PRADO	106
20.2.4. PORTFOLIO, PROGRAMME AND PROJECT MANAGEMENT MATURITY MODEL – OGC	106
20.2.5. KERZNER PROJECT MANAGEMENT MATURITY MODEL – HAROLD KERZNER	107
<u>20.3. OPM3</u>	<u>108</u>
20.3.1. ESTRUTURA DO MODELO.....	108
20.3.2. AVALIAÇÃO DA MATURIDADE	109
20.3.3. IMPLANTAÇÃO DO MODELO	110
<u>20.4 MMGP</u>	<u>112</u>
20.4.1. ESTRUTURA DO MODELO.....	112
20.4.2. AVALIAÇÃO DA MATURIDADE	113
20.4.3. IMPLANTAÇÃO DO MODELO	114
<u>20.5. KPMMM</u>	<u>114</u>
20.5.1. ESTRUTURA DO MODELO.....	114
20.5.2. AVALIAÇÃO DA MATURIDADE	116
20.5.3. IMPLANTAÇÃO DO MODELO	120
<u>20.6. UM ESTUDO DE CASO.....</u>	<u>121</u>
20.6.1. METODOLOGIA	122
20.6.2. RESULTADOS COLETADOS.....	122
20.6.3. PERFIL DOS PARTICIPANTES	123
20.6.4. SEGMENTAÇÃO POR NÍVEL DE MATURIDADE	125
20.6.5. SEGMENTAÇÃO POR PERCENTUAL DE ADERÊNCIA AOS NÍVEIS DE MATURIDADE	126
20.6.6. CONCLUSÃO	129
<u>20.7. ANÁLISE COMPARATIVA.....</u>	<u>129</u>

20.9. TÓPICOS DE PESQUISA 101

20.10. EXERCÍCIOS 101

REFERÊNCIAS..... 102

Além disso, no final do capítulo, será demonstrado um estudo de caso da avaliação da maturidade de empresas juniores no Brasil utilizando o MMGP, uma análise comparativa entre os principais modelos de maturidade apresentados, algumas sugestões de leitura para quem desejar se aprofundar em algum dos temas abordados, tópicos de pesquisa referentes ao tema e por fim exercícios para fixação do conteúdo trabalhado.

Comment [C656]: Disto,

Comment [C657]: ao final do capítulo

Antes de começar a abordar o tema da maturidade em gerenciamento de projetos é necessário definir alguns termos que serão frequentemente abordados durante todo este trabalho. Para isso é importante que o significado dos mesmos seja conhecido objetivando garantir o pleno entendimento de todo o contexto.

Comment [C658]: torna-se

Comment [C659]: capítulo

Comment [C660]: achei desnecessário este trecho

Comment [C661]:

Comment [C662R661]:

Comment [C663R662]: .

Comment [C664]: remover

Comment [C665]: De acordo...

Comment [C666]: Outra definição relevante...

A primeira e talvez a mais importante definição a ser citada seja a da palavra projeto que de acordo com o PMBOK [PMI 2005] é um empreendimento único, com início e fim definidos, que utiliza recursos limitados e é conduzido por pessoas, visando atingir metas e objetivos pré-definidos estabelecidos dentro de parâmetros de prazo, custo e qualidade.

Além da definição de projeto também é necessário entender o que é gerenciamento de projetos. Segundo o PMBOK [PMI 2005] gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas nas atividades do projeto a fim de atender

Comment [0667]: Estes conceitos não são apresentados em capítulos anteriores?

Maturidade pode ser definida como o desenvolvimento de sistemas e processos que são, por natureza, repetitivos e garantem uma alta probabilidade de que cada um deles seja um sucesso [Kerzner 2002]. Contudo, processos e sistemas repetitivos não são por si só, garantia de sucesso, apenas aumentam sua probabilidade. Assim, podemos observar que a maturidade em gestão de projetos está ligada a quão hábil uma organização está no exercício de gerenciar seus projetos [Prado 2004].

Atualmente existe uma necessidade de padronização dos processos de uma organização com o objetivo de elevar o nível de eficácia em gerenciamento de projetos possibilitando competir no acirrado mercado globalizado. Para isso, estão disponíveis cerca de 27 modelos de maturidade [Oliveira 2006] que indicam caminhos pelos quais a implementação de padrões pode tornar uma organização mais produtiva e competitiva.

O estudo da maturidade em gerenciamento, assim como o estudo do próprio gerenciamento de projetos, é assunto que vem sendo discutido recentemente, mas tem ocupado lugar de destaque [Leal 2008].

Muitos dos modelos de maturidade para gerenciamento de projetos apresentam a estrutura de cinco níveis proposta pelo CMM/CMMI, contudo algumas diferenças são observadas no conteúdo de cada nível. Dentre os modelos existentes podem ser destacados PMMM [Crawford 2002], MMGP [Prado 2004], P3M3 [OGC 2006], OPM3 [PMI 2003] e o KPMMM [Kerzner 2003], que serão apresentados a seguir.

Esse modelo foi concebido em 2003 e lançado no início de 2004 pelo PMI – *Project Management Institute* contando com a participação de aproximadamente 800 voluntários de mais de 35 países, inclusive do Brasil, na sua elaboração.

O modelo inclui a recomendação de 603 boas práticas, ou seja, as experiências vividas ao longo dos anos por diversas empresas e profissionais da área de gerência de projetos. Desse modo o modelo retrata uma trilha aparentemente segura e referenciada, capaz de orientar os gestores organizacionais nos seus investimentos e iniciativas de aprimoramento da operação de gestão de projetos.

Uma das questões bastante discutidas na literatura é a relação muitas vezes inexistente entre o planejamento estratégico e o planejamento de Sistemas de Informação, principalmente no que diz respeito à estratégia organizacional e os projetos da organização.

O OPM3 como outros modelos de maturidade organizacional procura reconhecer e sinalizar as fases de amadurecimento progressivo da organização a cada degrau alcançado. Ao

Comment [C668]: Maturidade, por sua vez...

Comment [C669]: .

Comment [C670]: remover

Comment [C671]: mas aumentam a probabilidade de sucesso.

Comment [C672]:

Comment [C673R672]: Além destes conceitos, essências para entendimento do restante deste capítulo, atualmente torna-se nítida a necessidade...

Comment [C674]: Colocar por extensor entre parenteses

Comment [C675]: nas organizações

Comment [C676]: em particular...

Comment [C677]: remover itálico

Comment [C678]: colocar por extensor e entre parenteses

Comment [C679]: , muitas vezes inexistente,

Comment [C680]: , como outros modelos de maturidade organizacional,

O modelo de maturidade PMMM (*Project Management Maturity Model*) foi concebido, em 2002, pelo Center for Business Practices, área que cuida das pesquisas da PM Solutions, organização voltada para consultoria e treinamentos em gerenciamento de projetos, leva em consideração as 9 áreas de conhecimento da gestão de projetos descritas em [PMI 2005], que são: integração, escopo, tempo, custo, qualidade, recursos humanos, comunicações, riscos e aquisições, além disso, a avaliação do grau de maturidade é baseada em cinco níveis cujas características [Prado 2004] serão detalhadas melhor abaixo:

- **Nível 1 – Processos iniciais**

- Análise informal da performance dos projetos;

Comment [C681]: Tb extenso e entre parenteses

Comment [C682]: melhor detalhadas

Comment [C683]:

Comment [C684R683]: colocar estas partes em negrito para destacar

Comment [C685]: itálico????

- São ativados processos para melhorar a eficiência da performance dos projetos;

Comment [C686]: itálico????

A Figura 20.1 demonstra a proposta da PM Solutions que foi criar um modelo de maturidade incorporando os 5 níveis de maturidade evolucionais existentes no *Capability*

Comment [C687]: extenso

20.2.3. Modelo de Maturidade em Gerenciamento de Projetos – Darci Prado

Ambos os modelos são baseados em cinco níveis de maturidade e possuem um sistema de avaliação similar onde se deve responder a um questionário contendo 40 perguntas de múltipla escolha e a partir das respostas obtidas é calculado o nível de maturidade seja ele setorial ou corporativo.

O modelo de maturidade P3M3 (*Portfólio, Programme and Project Management Maturity Model*) foi desenvolvido pela OGC, *Office of Government Commerce*, a Secretaria de Comércio Governamental do Reino Unido, mesma instituição que desenvolveu o PRINCE2. A versão inicial do P3M3 foi lançada em fevereiro de 2006, porém desde junho de 2008 está disponível a versão 2.0 do modelo que atualmente está em fase de consulta pública, sujeita a comentários e sugestões de usuários e especialistas.

O modelo P3M3 pode ser usado como base para a melhoria dos processos de gerenciamento de projetos, programas e portfólio. Ele está estruturado em 5 níveis de maturidade que são:

Comment [C689]: Talvez fosse necessário escrever conforme os demais modelos... tem pouco conteúdo em relação aos demais não?

Comment [C690]: extenso

Comment [C691]:

Comment [C692R691]: .

Comment [C693]: remover

Comment [C694]: Diante disto, a partir ...

Comment [C695]: Entre parenteses não?

Comment [C696]: extenso



Comment [C699]: Diminuir tamanho da figura para caber ainda na pagina anterior

- Domínios de abrangência, sobre os quais se desenvolvem as indicações e recomendações de amadurecimento organizacional. Os domínios mencionados no Modelo OPM3 referem-se a projetos, programas e portfólio;
- Estágios de amadurecimento dos processos organizacionais, associados às fases do modelo de melhoria contínua de processos. Os estágios de amadurecimento são descritos no Modelo OPM3 como Estágio de Padronização, de Mensuração, de Controle e de Melhoria Contínua.

Comment [C700]: :

Cada domínio do gerenciamento de projeto organizacional (Projeto, Programa e Portfólio) apresenta estágios de melhoria de processo (Padronização, Medição, Controle e Melhoria Contínua). Da mesma forma, as capacidades estão também mapeadas para cada um dos cinco grupos de processos em gerenciamento de projeto (iniciação, planejamento, execução, controle e encerramento). O desenvolvimento de capacidades em cada um dos grupos de processos descritos no PMBOK irá auxiliar a empresa na implementação dos planos

Comment [C701]: :

Figura 20.3 demonstra o relacionamento entre os domínios, os estágios e os grupos de processos especificados no modelo.

Este questionário é composto por 151 perguntas abrangendo os domínios de Projetos, Programas e Portfólios. Inicialmente estas perguntas foram divididas de acordo com o domínio que avaliavam (70 perguntas para Projeto, 38 perguntas para Programas e 43 perguntas para Portfólio).

Comment [C702]: extenso

Comment [C703]:

Comment [C704R703]: extenso

Comment [C705]: extenso

Comment [C706]: extenso

Depois de associar o valor correspondente a cada uma das respostas dever-se-ia aplicar a fórmula a seguir para a partir da mesma extrair o valor referente ao nível de maturidade alcançado

Comment [C711]: , a partir da mesma, extrair

- Interesses pessoais **vêm** à frente de interesses corporativos;

Comment [C712]: vêm

A avaliação é realizada através de um questionário que contém 183 questões sendo divididas em 80, 20, 42, 25 e 16 questões, respectivamente para os níveis de 1 a 5.

O questionário referente ao nível de maturidade 1 contém 80 perguntas de múltipla escolha com cinco alternativas de resposta para cada questão, sendo que apenas uma delas é a resposta correta. O conteúdo destas questões são os princípios básicos do gerenciamento de projetos contidos no PMBOK [PMI 2005]. Apesar do PMBOK ser dividido em 9 áreas de conhecimento, para simplificar o questionário as áreas de conhecimento, gerenciamento de escopo e de integração foram integradas como uma única área. Sendo assim as o questionário foi dividido em 10 perguntas para cada uma das seguintes áreas de acordo com a seguinte tabela:

Comment [C713]:

Comment [C714R713]: numeros extenso

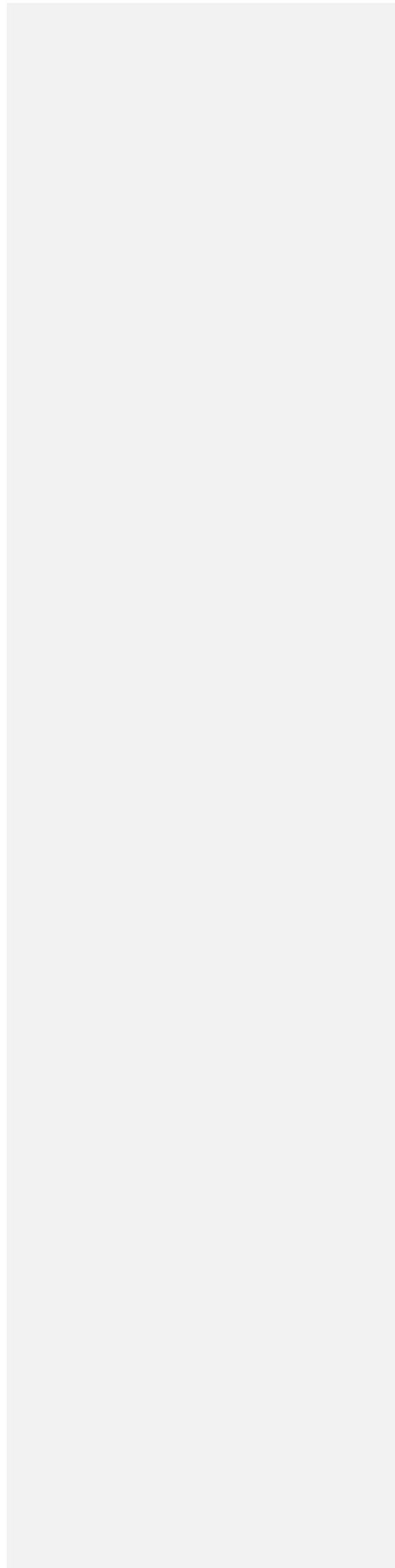
Comment [C715]: numero extenso

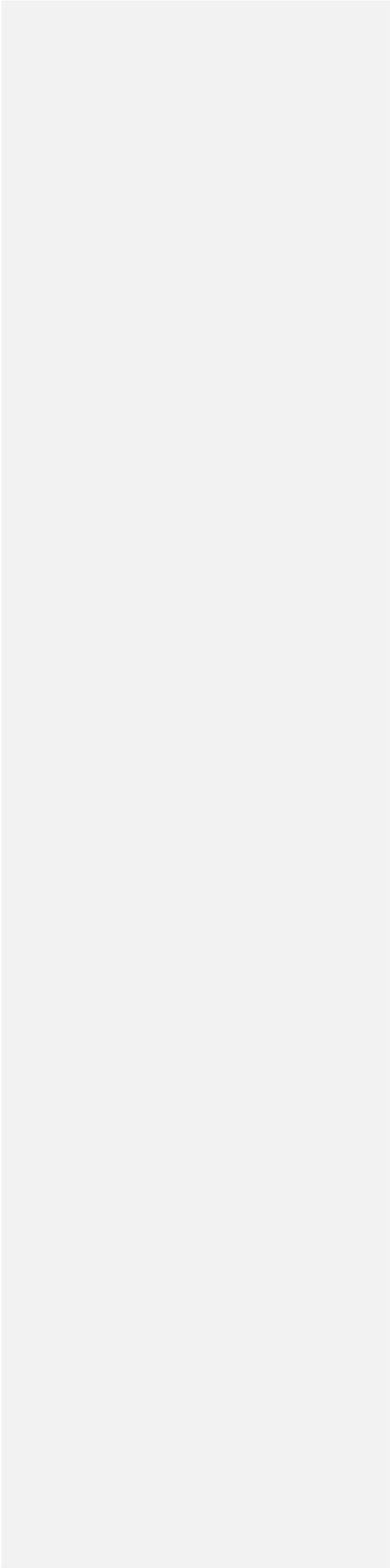
Os pontos devem ser contabilizados da seguinte maneira após a conclusão do questionário: deve-se somar 10 pontos a cada resposta correta e nenhum ponto em caso de

respostas erradas. O somatório dos pontos deve ser feito de acordo com a divisão das perguntas por área, sendo assim o intervalo de pontos por área varia de 0 a 100.

- Se sua organização obter menos de 60 pontos em todas as categorias, existe uma deficiência. Para pontuações inferiores a 30 em qualquer uma das categorias é necessário um treinamento rigoroso sobre os princípios básicos de gerenciamento de projetos. A organização aparenta estar altamente imatura em gerenciamento de projetos;

Comment [C716]: obtiver





Para fazer o levantamento da Avaliação da Maturidade nas Empresas Juniores brasileiras foi realizada uma pesquisa referente aos modelos de maturidade mais utilizados hoje em dia para identificarmos qual se adaptaria melhor ao estudo proposto e foi identificada

Comment [C717]: .

que o modelo mais adequado para esta pesquisa seria o Modelo de Maturidade em Gerenciamento de Projetos – MMGP proposto por Darci Prado [Prado 2004].

O questionário continha uma área inicial para cadastro dos respondentes e para futura segmentação e análise dos dados de acordo com as necessidades composto por oito questões referentes à identificação da empresa júnior (nome, área de atuação, cidade, estado e instituição de ensino superior a qual está vinculada) e também do respondente (cargo ocupado, nome e e-mail para contato).

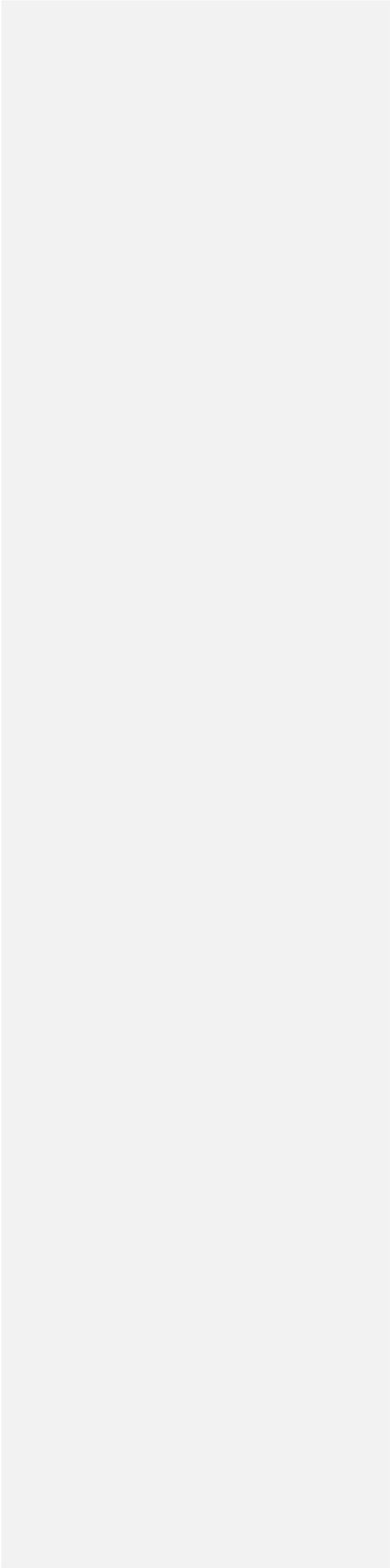
Comment [C718]: Remover espaço adicional

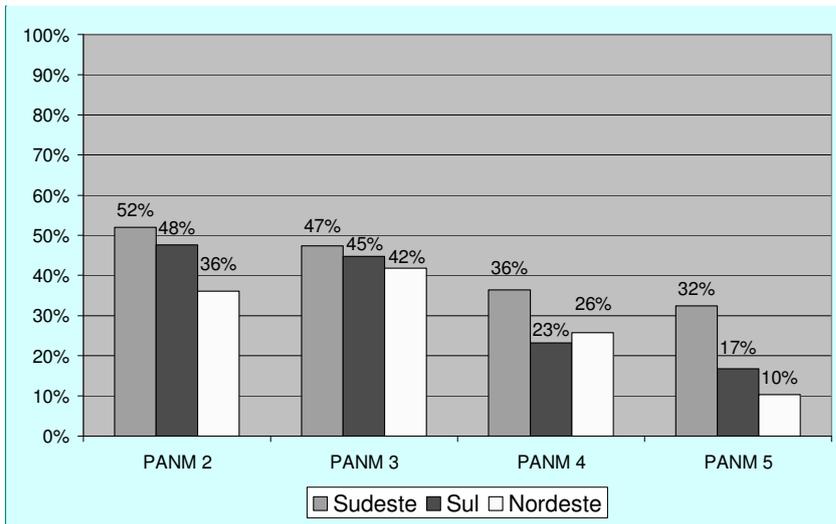
Destes 108 convites enviados foram coletados 33 questionários respondidos completamente e 2 respondidos parcialmente, sendo estes descartados por inconformidade com os padrões. Dentre os 33 questionários válidos temos representação de 8 estados, sendo 4 da região Nordeste (BA, PB, PE e SE), 2 da região sudeste (MG e RJ) e mais 2 da região Sul (PR e SC). Além disso, tentamos abordar 36 Instituições de Ensino Superior que oferecem 55 cursos de graduação. destes contemplamos nos resultados 25 cursos de graduação em 18

empresas juniores e suas respectivas áreas de atuação e instituição de ensino superior a qual estão vinculadas.

Detalhando-se um pouco mais os dados percebemos que o estado que possui o maior nível de Maturidade é Minas Gerais, seguido pelo Rio de Janeiro, ambos representantes da região sudeste, o Paraná vem em seguida representando a região Sul e por fim temos Bahia, Pernambuco e Paraíba representando a região Nordeste. Os estados de Santa Catarina e Sergipe não estão presentes no gráfico acima, pois foram consideradas apenas unidades federativas com pelo menos três EJ's respondentes enquanto que estes possuíram apenas uma EJ participante.

Comment [C719]: , com pelo menos três EJ's respondentes.





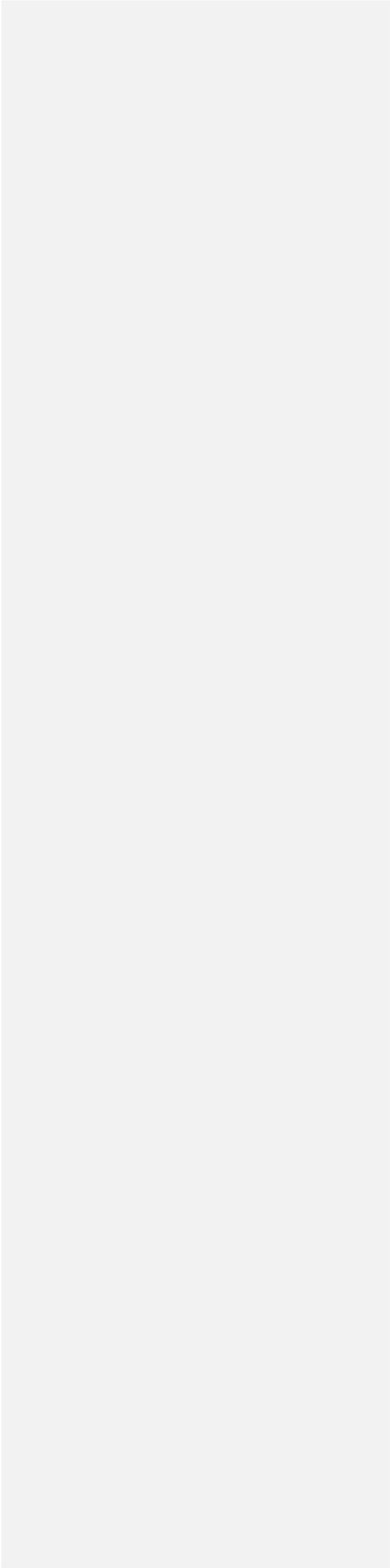
Comment [C720]:

Comment [C721R720]: Reduzir tamanho do gráfico para colocá-lo junto ao seu texto

Podemos obter dados importantes através da comparação com a pesquisa de Archibald e Prado [Prado, Archibald 2006] a qual indica que o nível de maturidade das Empresas Juniores brasileiras se assemelha bastante com os níveis das organizações privadas, do terceiro setor e do governo que possuem administração indireta. Isso gera uma confiabilidade maior as empresas juniores perante a sociedade, visto que isto equipara este setor aos outros.

Comment [C722]: Aos níveis

			Padronização Organizacional			



			Institucionais			

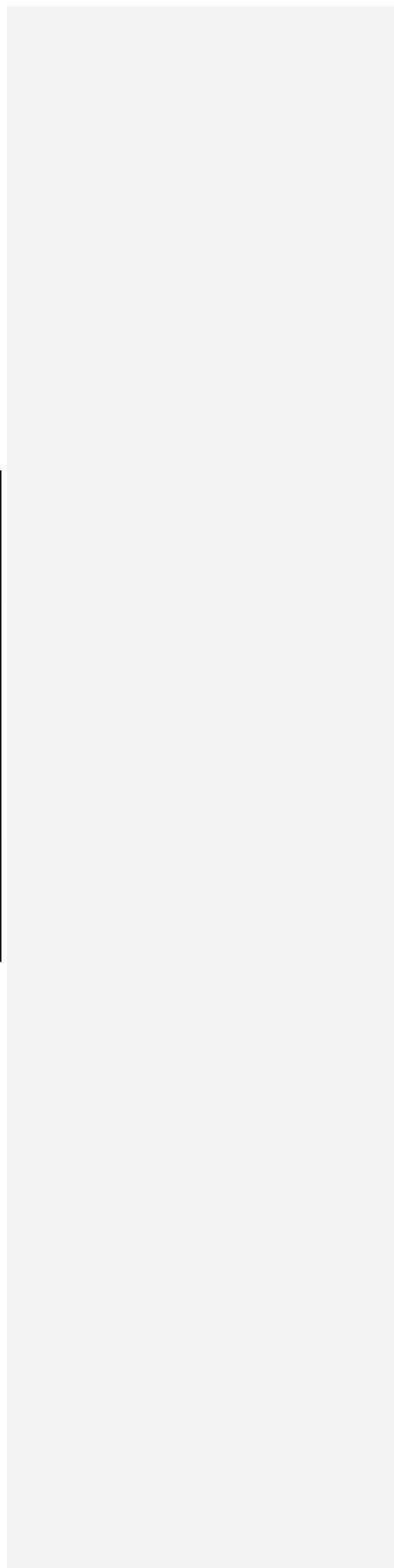


Tabela 20.9:Dados Gerais dos Modelos de Gestão de Projetos [Adaptado de Costa 2009].

Comment [C723]:
Comment [C724R723]: Tabela incompleta

20.9. Tópicos de Pesquisa

Comment [C725]:

Comment [C726R725]: Colocar outros tópicos

<http://www.maturityresearch.com/2006/downloads/RelatorioFinal_Completo_MPC_M_2006.pdf> Acesso em: 15 dez. 2007

21. GOVERNANÇA EM TIC	4
21.1 GESTÃO EM TIC.....	4
21.1.1 RELEVÂNCIA E EVOLUÇÃO DO PAPEL DA TIC NAS ORGANIZAÇÕES	6
21.1.2 DA GESTÃO À GOVERNANÇA EM TIC	9
21.2 MODELOS DE GESTÃO EM TIC	12
21.2.1 COBIT.....	12
21.2.2 ITIL.....	12
21.2.3 BSC	12
21.2.4 IT FLEX	13
21.2.5 COSO	14
21.2.6 ISO/IEC 20000.....	15
21.2.7 VAL IT.....	15
21.2.8 CMMI SOB A PERSPECTIVA DE GOVERNANÇA DE TI.....	16
21.3 ITIL	16
21.3.1 DEFINIÇÃO	16
21.3.2 HISTÓRICO	16
21.3.3 REGULAMENTAÇÃO DO ITIL	17
21.3.3.1 DIREITOS AUTORAIS	18
21.3.3.2 CERTIFICAÇÕES / TREINAMENTOS	18
21.3.3.3 PUBLICAÇÃO DE CONTEÚDOS OFICIAIS	18
21.3.3.4 FÓRUM DE FOMENTO (ITSMF).....	18
21.3.4 ESTRUTURA DO ITIL	19
• SERVICE STRATEGY (ESTRATÉGIA DE SERVIÇOS).....	19
• SERVICE DESIGN (PLANEJAMENTO DE SERVIÇOS)	19
• SERVICE TRANSITION (TRANSIÇÃO DE SERVIÇOS).....	19
• SERVICE OPERATION (OPERAÇÃO DE SERVIÇOS).....	19
• CONTINUAL SERVICE IMPROVEMENT (APRIMORAMENTO CONTÍNUO DE SERVIÇOS)	19
21.3.5 O QUE NÃO É ITIL.....	20
21.3.6 FRONTEIRAS COM OUTROS MODELOS E LIMITAÇÕES	21
21.3.7 PONTO DE PARTIDA	22
21.3.8 COMENTÁRIOS SOBRE PRÁTICAS DE SUCESSO.....	23
21.3.9 PÚBLICO ALVO	24
21.3.10 UTILIZAÇÃO DO ITIL.....	25
21.4 COBIT	26
21.4.1 DEFINIÇÃO	26

21.4.2	HISTÓRICO	27
21.4.3	REGULAMENTAÇÃO DO COBIT	28
21.4.3.1	CERTIFICAÇÕES / TREINAMENTOS	28
21.4.3.2	DIREITOS AUTORAIS	28
21.4.3.3	PUBLICAÇÃO DE CONTEÚDOS OFICIAIS	28
21.4.3.4	FÓRUM DE FOMENTO (ISACA)	29
21.4.4	ESTRUTURA DO COBIT	29
21.4.4.1	PRIMEIRA DIMENSÃO DO CUBO – PROCESSOS DE TI.....	29
21.4.4.2	SEGUNDA DIMENSÃO DO CUBO – CRITÉRIOS DE INFORMAÇÃO	32
21.4.4.3	TERCEIRA DIMENSÃO DO CUBO – RECURSOS DE TI.....	33
21.4.5	NÃO É COBIT	34
21.4.6	FRONTEIRAS COM OUTROS MODELOS.....	34
21.4.7	PONTO DE PARTIDA	35
21.4.8	COMENTÁRIOS SOBRE PRÁTICAS DE SUCESSO.....	36
21.4.9	PÚBLICO ALVO	37
21.4.10	UTILIZAÇÃO DO COBIT.....	37
21.5	INICIATIVAS DE INTEGRAÇÃO DOS PRINCIPAIS MODELOS	38
21.6	IMPLANTAÇÃO DE MODELOS DE GESTÃO	39
21.7	TÓPICOS DE PESQUISA	41
21.8	SUGESTÕES DE LEITURA	43
21.9	EXERCÍCIOS.....	45
21.10	REFERÊNCIAS	47

Comment [F727]: Fonte diferente

Capítulo

Governança em TIC

Comment [WU728]: Texto deveria estar alinhado a esquerda e com fonte tamanho 20 e Times New Roman

O objetivo deste Capítulo é apresentar os principais conceitos de governança em TIC – Tecnologias da Informação e Comunicação, sua evolução e origens, assim como, apresentar os modelos mais difundidos nessa área, enfatizando os modelos ITIL – Information Technology Infrastructure Library e COBIT - Control Objectives for Information and related Technology, além de conceituar alguns outros menos difundidos para ampliação dos horizontes do leitor. São apresentados os conceitos mais importantes da Gestão de TIC, os modelos existentes nesta área, as iniciativas de integração dos principais modelos e como se dá a implantação de modelos de gestão.

Comment [WU729]: Verificar se o texto que introduz o capítulo vai ficar escrito em "itálico" ou "normal"

Gestão em TIC

Figura 21.1.1– Linha do Tempo da TIC. Fonte: Elaboração própria.

Comment [WU730]: Acho que basta colocar figura 21.1, não sei se precisa colocar que a figura é de elaboração dos autores.

Tabela 21.1.1 – Custo horário médio de interrupção de Serviços de TIC por natureza do negócio. FONTE: [MAGALHÃES 2007].

Comment [WU731]: Quando a legenda da tabela ultrapassar uma linha o alinhamento deve ser feito da seguinte maneira: Recuo à esquerda e a direita de 0,8 cm, alinhamento: justificado, espaçamento de 6 pt antes e depois além de fonte: Helvetica

Figura 21.1.2– Evolução da Gestão dos Departamentos de TIC. Fonte: Adaptado de [FERNÁNDEZ 2008].

Comment [WU732]: Idem comentário 4

Ênfase demasiada em TIC: um dos erros comuns que se cometem ao implantar ferramentas de Governança de TIC é justamente analisá-las sobre o ponto de vista tecnológico.

Necessidade inerente de estruturar: os departamentos de TIC estão acostumados a estruturar os componentes que fazem parte do ambiente de TIC; o problema surge quando esta estruturação termina por criar “forçadamente” responsabilidades e perfis, estruturas hierárquicas rígidas, definições de processos excessivamente formalizados, todos eles dependentes de acordos estáticos de nível de serviço. Neste ponto, o excesso de formalismo pode transformar toda a estruturação em um modelo rígido e inútil.

Não se consideram as pessoas: numa organização são as pessoas que efetivamente efetuam, controlam e decidem sobre processos, e são as pessoas de negócio que, com suas decisões,

Formatted: Indent: Left: 0,73", Bulleted +
Level: 1 + Aligned at: 0,94" + Indent at:
1,19"

procuram a geração de valor na empresa. Contudo, todas as ferramentas de Governança de TIC atuais seguem centrando-se nas estruturas e nos processos. É necessário que existam mecanismos eficazes que fomentem a relação, a comunicação e a colaboração entre as pessoas e a organização, no contexto das estruturas e dos processos.

- A liderança do CIO²⁷: tradicionalmente a figura do CIO tem sido a de se apresentar como o “paladino das causas do departamento de TIC”, procurando defender os investimentos em infraestrutura de TIC, e atuando no máximo em nível tático. É necessário, contudo, que esta figura se reposicione estrategicamente na organização, respondendo diretamente ao CEO²⁸ e apoiando-o no processo de decisão estratégica da organização. Para que isso aconteça, no entanto, é necessário que a TIC deixe de ser um centro de altos custos da organização e passe a atuar na camada estratégica do negócio, como setor de inovação e diferencial competitivo.

À medida que as organizações começaram a reconhecer a sua dependência crescente na TIC para conseguirem satisfazer os objetivos do negócio, caminhando ao encontro às necessidades da organização, muitos autores determinaram como fundamental a garantia de uma maior qualidade dos serviços de TIC, e a sua gestão efetiva [MAGALHÃES 2007].

Comment [WU733]: Acho que a concordância correta seria “caminhando de encontro às ” ou então “caminhando ao encontro das”

²⁷ CIO – Chief Information Officer

²⁸ CEO – Chief Executive Officer

Finalmente poderíamos definir Governança de TIC como o alinhamento estratégico de TIC com o negócio de forma que se obtenha o máximo valor deste através do desenvolvimento e manutenção de controles efetivos de TIC orientados ao controle de custos, gestão do retorno dos investimentos

relacionados e gestão dos riscos associados [WEILL&ROSS 2005]. A Figura 21.1.3 a seguir ilustra bem a relação entre as diversas áreas mencionadas.

Figura 21.1.3 – Diagrama de Interrelação entre as áreas. Fonte: Elaboração própria.

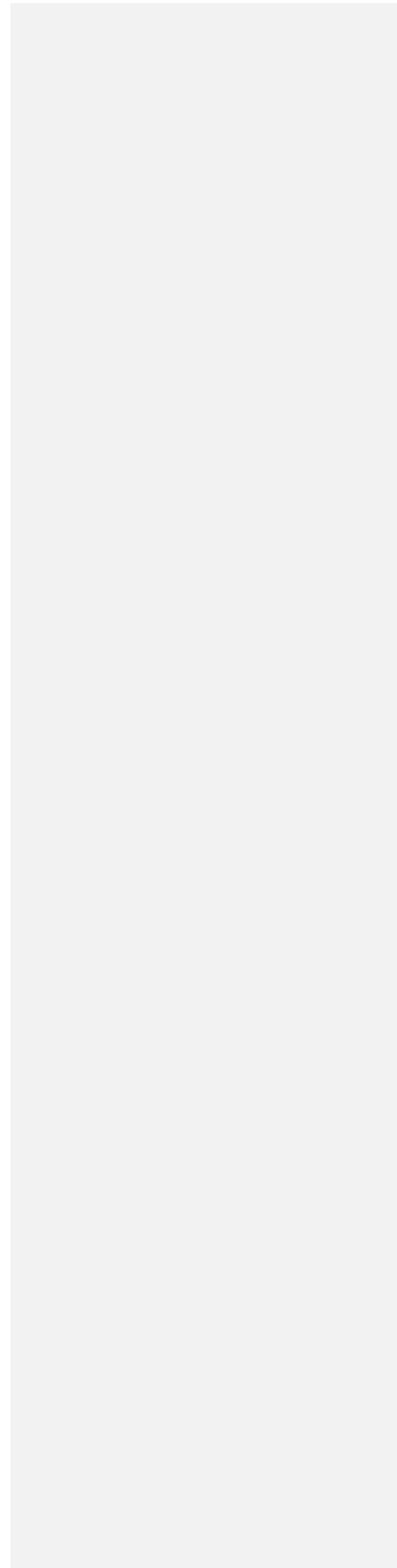
Comment [F734]: É necessário?

O COBIT – *Control Objectives for Information and related Technology* é um framework de governança de TI apoiado por um conjunto de ferramentas que permite aos gestores fazer a ponte entre as exigências de controle, questões técnicas e riscos do negócio. Ele permite o desenvolvimento de políticas claras e boas práticas de controle de TI em toda a organização. Além disso, o COBIT enfatiza a conformidade

regulamentar, ajuda as organizações a aumentar o valor obtido através da TI, permite o alinhamento e simplifica sua implementação.

Recentemente vem se usando também a descrição *Control Objectives of Sarbanes Oxley*, para a sigla COSO. Como o *Internal Control – Integrated Framework* é um modelo de trabalho muito genérico,

com visão de auditoria, muitas organizações usam o COBIT (*Control Objectives for Information and Related Technology*) para aplicar o COSO. Na prática, o que acontece é que empresas adotam o COSO de forma geral, para controles internos, principalmente financeiros. A área de TI, por sua vez, adota o COBIT, como guarda-chuva para diversas metodologias e melhores práticas indicadas para tecnologia da informação.



O *Office of Government Commerce*, cujo web site pode ser visto em [OGC 2009], é o “detentor dos direitos” do ITIL. A missão do OGC é trabalhar com o setor público como catalisador para atingir maior eficiência, aumentar valor nas atividades comerciais, e melhorar o sucesso no fornecimento de programas

e projetos. Quando se olha para o OGC, pode-se verificar que a abrangência das suas preocupações é muito maior que apenas a melhoria da TI, estendendo estas a outras áreas diversas.

Estrutura do ITIL

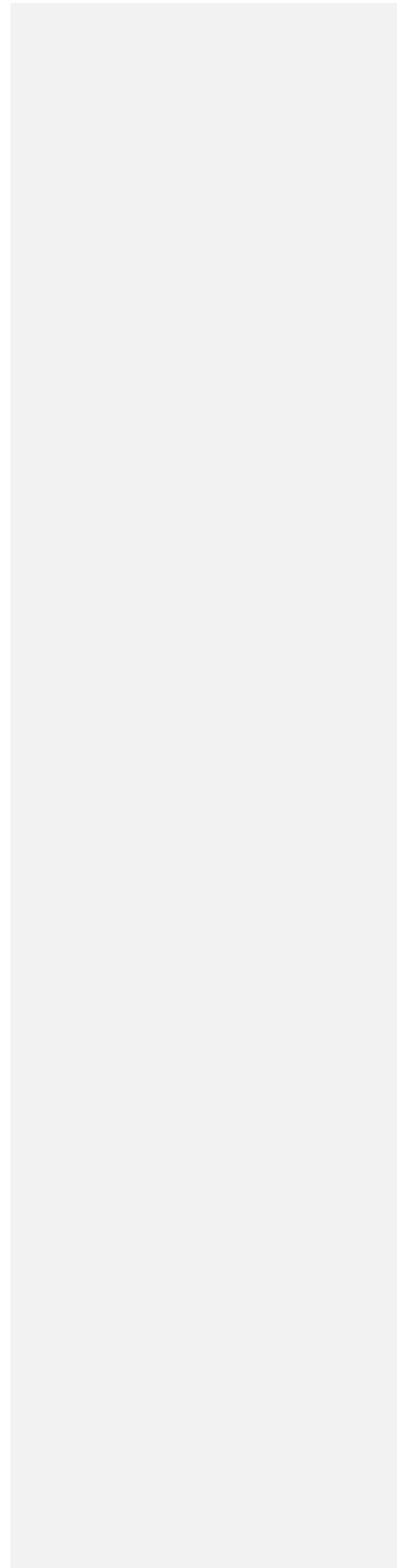
Comment [F735]: Acho que foi feita uma análise geral e crítica sobre o modelo, mas senti falta do conteúdo propriamente dito dos livros, principalmente em relação aos processos de Gerenciamento de Incidentes, Problemas, Configuração, etc, que são comentados adiante num ranking de preferencia, mas ficou faltando o conteúdo. A introdução ficou bastante grande, acho que poderia diminuir os aspectos gerais e focar mais no modelo em si. É uma sugestão apenas.

É um guia para criação e manutenção de políticas e arquiteturas para o planejamento de serviços. Esse livro abrange o ciclo de vida dos serviços, papéis e responsabilidades, objetivos e elementos dos serviços,

a seleção do modelo do serviço, o modelo de custo, riscos e benefícios, implementação e fatores de sucesso.

A principal vantagem da aproximação do ITIL às “melhores práticas” é que os processos descritos são genéricos, aplicam-se independentemente da tecnologia, plataforma, tipo ou tamanho do negócio envolvido. Quase todas as organizações das TI de qualquer tamanho têm um “help desk”, um método de lidar com problemas ou mudanças, alguma compreensão de gestão de configuração, níveis de serviço de acordo com os clientes, uma maneira de lidar com problemas de capacidade e disponibilidade e uma forma de plano de contingência. O foco primário da metodologia ITIL é possibilitar que área de TI seja mais efetiva e proativa, satisfazendo assim clientes e usuários [ITSMF 2008].

Comment [F736]: É isso mesmo, ou seria níveis de acordo de serviço?



- ITIL não é uma metodologia para implementar processos de Gestão de Serviços de TI – é um *framework* flexível que permite adaptar-se para ir ao encontro das necessidades específicas;

Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,75" + Indent at: 1"

- ISO/IEC 20.000 - é a norma ISO para certificação de empresas no Gerenciamento de Serviços de TI, com base nas melhores práticas da ITIL.

Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,75" + Indent at: 1"

- IBM PRM-IT – É o *IBM Process Reference Model for IT* (PRM-IT), outro modelo proprietário baseado na versão 2 do ITIL [IBM PRM-IT 2004].

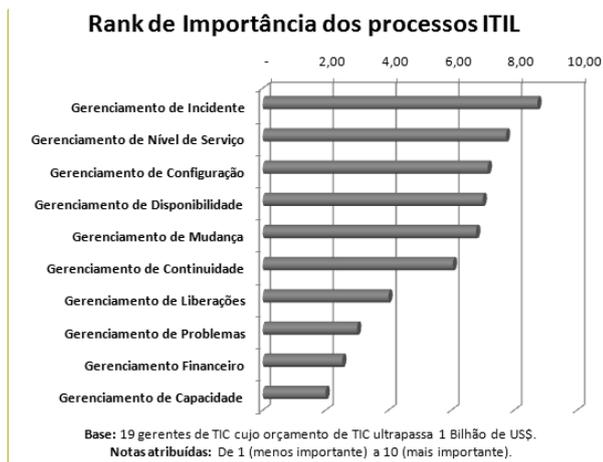
- **Os padrões do ITIL são difíceis de implementar** – O ITIL descreve o “quê” mas não o “como” do fornecimento de serviços. O ITIL não pode ser implantado exclusivamente através da leitura de seus livros, em sua versão atual.

Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,75" + Indent at: 1"

- **ITIL não é capaz de mapear os processos de negócio nos processos de TI** – no passado os vendedores de produtos e serviço “ITIL-compliant” teriam prometido aos seus compradores justamente isso.

Entre Novembro de 2004 e Fevereiro de 2005, uma pesquisa de sondagem foi realizada pela Forrester [MENDEL & PARKER 2005], em que foram entrevistadas 19 empresas que passaram pela implantação

de ITIL. Foi pedido às empresas para atribuírem um valor aos processos de ITIL, em termos de importância percebida e valor, identificados durante o processo. Os resultados desta sondagem estão ilustrados na Figura 21.3.3:



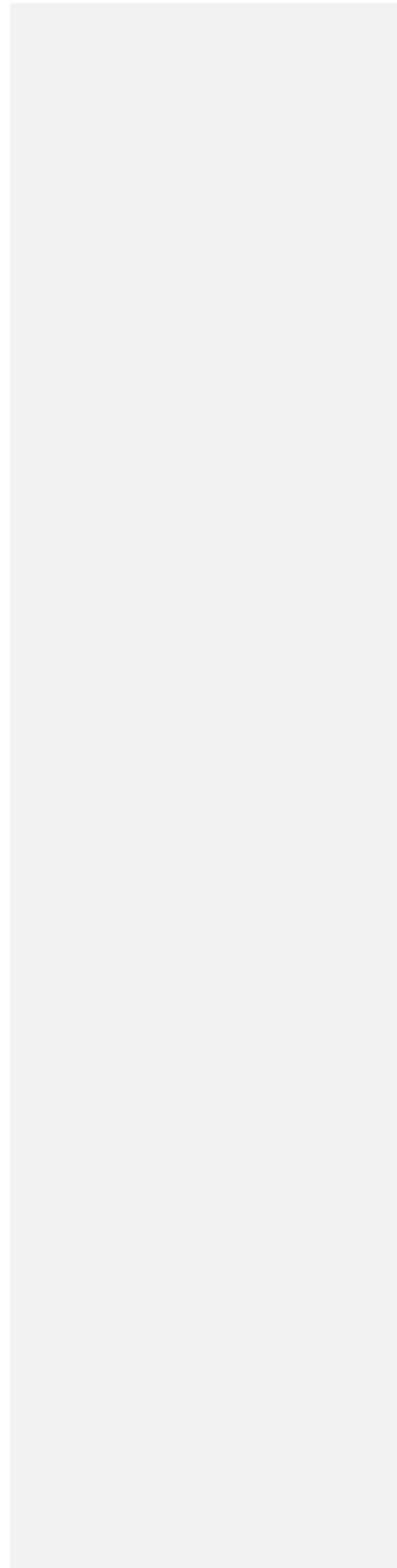
Comment [F737]: A figura mostra "Rank" não seria ranking?

- **Não fazer tudo de uma só vez:** o ITIL é bastante amplo. Adotar a implantação de muitos de seus processos simultaneamente é um risco que na maioria dos casos não compensa para a organização. Sugere-se escolher alguns poucos processos para iniciar e acelerar depois que a organização assimile melhor o modelo.
- **Pensar na avaliação desde o princípio:** uma das premissas do ITIL é melhorar a qualidade dos processos e serviços de TIC. Não é recomendável começar um processo de implantação

Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,75" + Indent at: 1"

sem se definir um processo de avaliação/medição eficaz. Estas avaliações devem ser realizadas no início do processo e na medida em que as mudanças forem ocorrendo. Definir critérios de qualidade claros e mensuráveis para os serviços é essencial para o sucesso do processo.

- **Priorizar a implantação de Gestão de Incidentes:** o gerenciamento da disponibilidade dos serviços é o coração do Gerenciamento de Serviços e foco central do ITIL, bem como a parte mais visível de toda a cadeia de governança de TIC. Iniciativas que promovam o tratamento de incidentes no contexto da organização é um bom começo.



Segundo o ISACA - *Information Systems Audit and Control Association*, a missão do COBIT é “Pesquisar, desenvolver, publicar e promover um conjunto de objetivos de controle para tecnologia que

seja embasado, atual, internacional e aceito em geral para o uso do dia-a-dia de gerentes de negócio e auditores” [ITGI 2007]. Para tanto, o COBIT trabalha principalmente dentro do seguinte conjunto de atividades:

10. ● Alinhamento da TI com o negócio da empresa;

Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,25" + Indent at: 0,5"

j) Responsabilidade corporativa: trata-se de pensar e agir pela perenidade da organização, com responsabilidade social e ambiental;

Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,25" + Indent at: 0,5"

m) ● Transparência: relacionado ao desejo de informar.

Formatted: Justified, Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,25" + Indent at: 0,5", Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers

		Melhoria dos controles para assegurar a segurança e disponibilidade dos ativos de

		TI na Organização

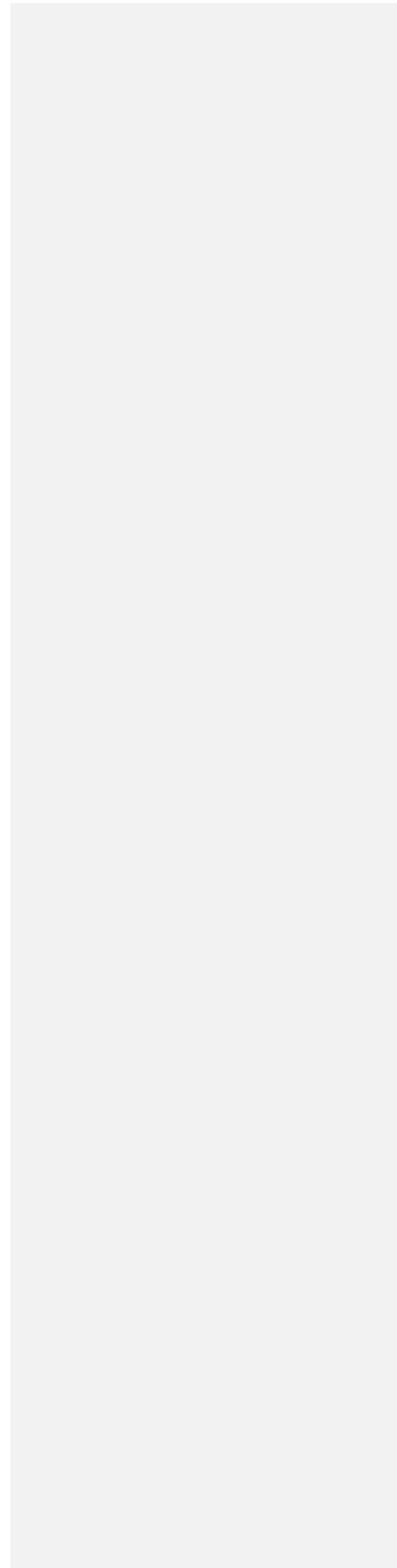
- f) Sumário executivo - Consiste em uma visão executiva que detalha os conceitos e princípios-chaves do COBIT.

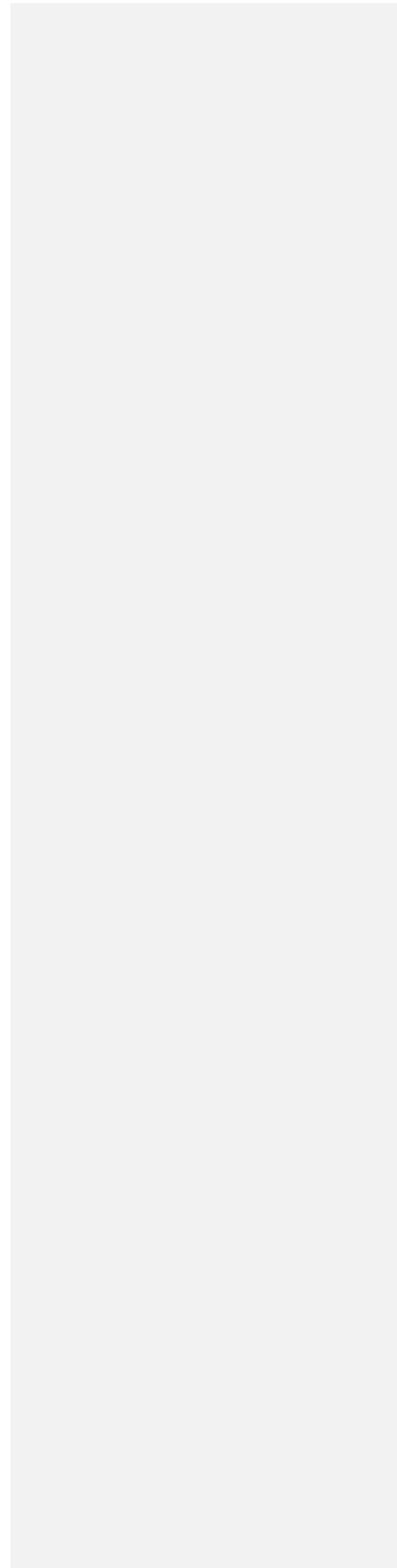
Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,25" + Indent at: 0,5"

| k) Ferramentas de implementação - Auxiliam na auto-avaliação da TI e na implementação, além das questões mais frequentes e estudos de casos.

Os processos do COBIT são agrupados nos seguintes domínios:

Formatted: Indent: First line: 0,5"





	Processos em aperfeiçoamentos, já fornecem as boas práticas. Mas faltam ferramentas de automação.

Comment [WU738]: Acho que o certo é "aperfeiçoamento"

2 Comprometimento e conscientização da alta administração e dos gestores;

Formatted: Indent: Left: 0,54", Bulleted + Level: 1 + Aligned at: 0,25" + Indent at: 0,5"

7 Possuir uma estrutura de tomada de decisão adequada aos objetivos estratégicos da organização;

Formatted: Justified, Indent: Left: 0,54", Space After: 0 pt, Bulleted + Level: 1 + Aligned at: 0,25" + Indent at: 0,5", Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers

12. Atender as exigências legais, regulatórias e os conflitos entre as próprias leis/regulamentos e também com as estratégias da organização.

Para que os modelos ITIL e COBIT sejam utilizados conjuntamente em um processo de implantação de Governança de TI é necessário fazer um mapeamento de correspondência entre os processos de ambos, resultando em uma matriz genérica que resulta no relacionamento desses dois

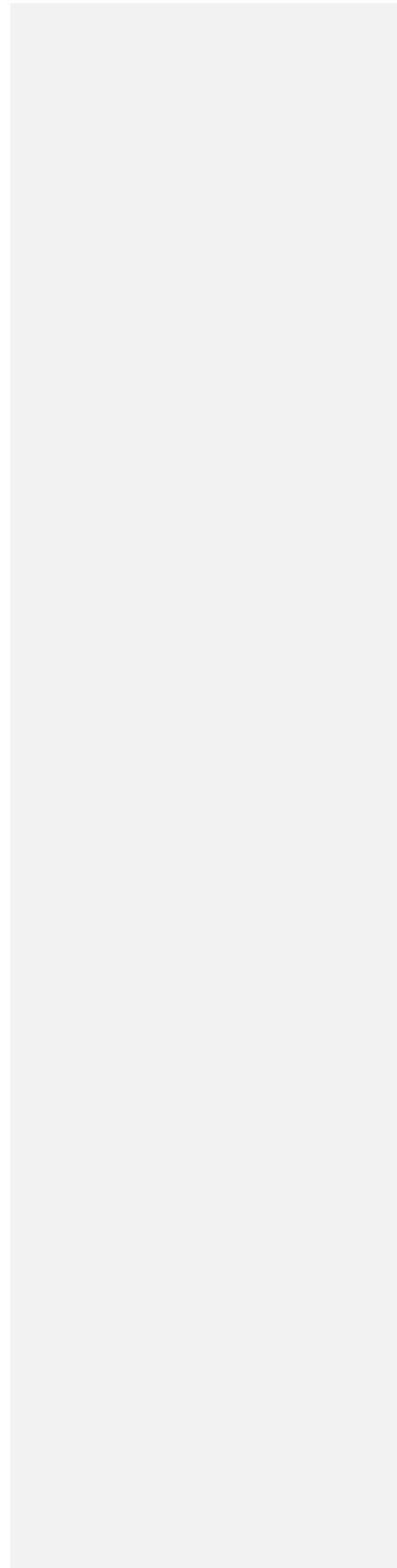
frameworks, onde em cada livro da biblioteca do ITIL existem objetivos de controle do COBIT que são aplicáveis ao processo do ITIL. Essa matriz pode ser visualizada na Tabela 21.5.1 [SODRÉ & SOUZA 2007].

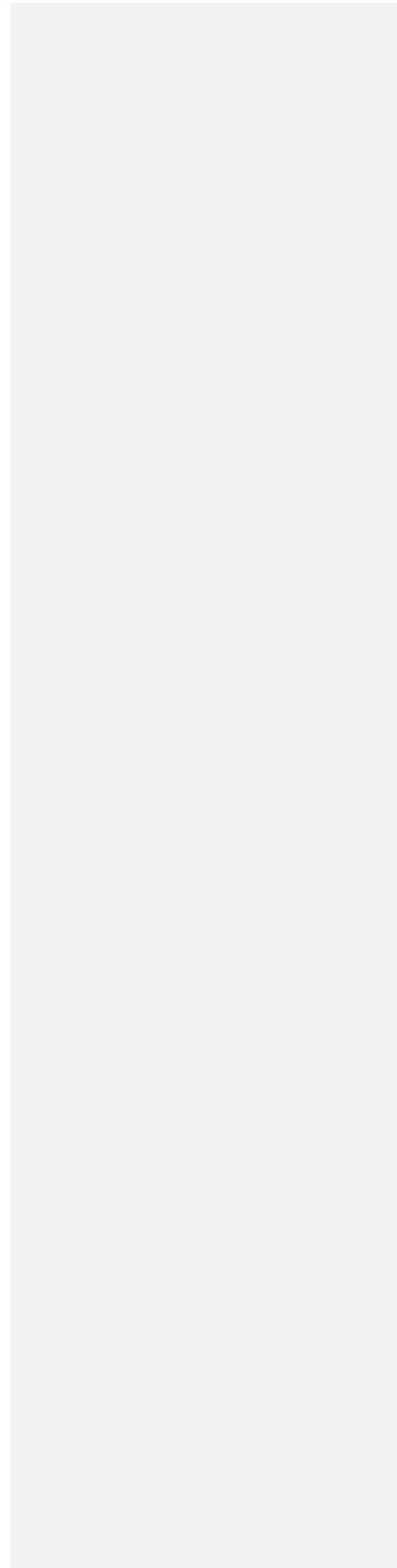
A implantação de governança de TIC não segue um método estático, por isso, esta seção tem o objetivo de ajudar na compreensão dos aspectos básicos de uma implantação. Com este objetivo

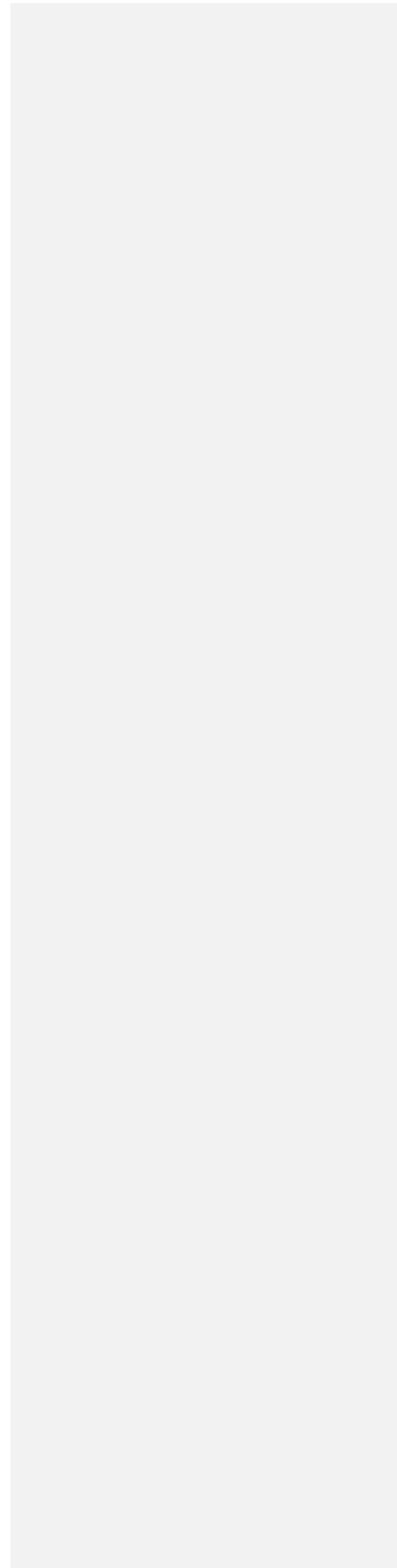
analisamos a forma como foram realizadas duas implantações de acordo com o relato dos autores das referências [TECHREPUBLIC 2002], [TECHREPUBLIC 2003].

- Governança Ágil em TIC é um tema extremamente recente e pouco explorado. Há até um movimento com o objetivo de fomentar a Governança Ágil de TIC nas organizações. Pode-se encontrar abundante material a respeito nas referências abaixo:

Comment [F739]: Vai ficar em negrito mesmo?





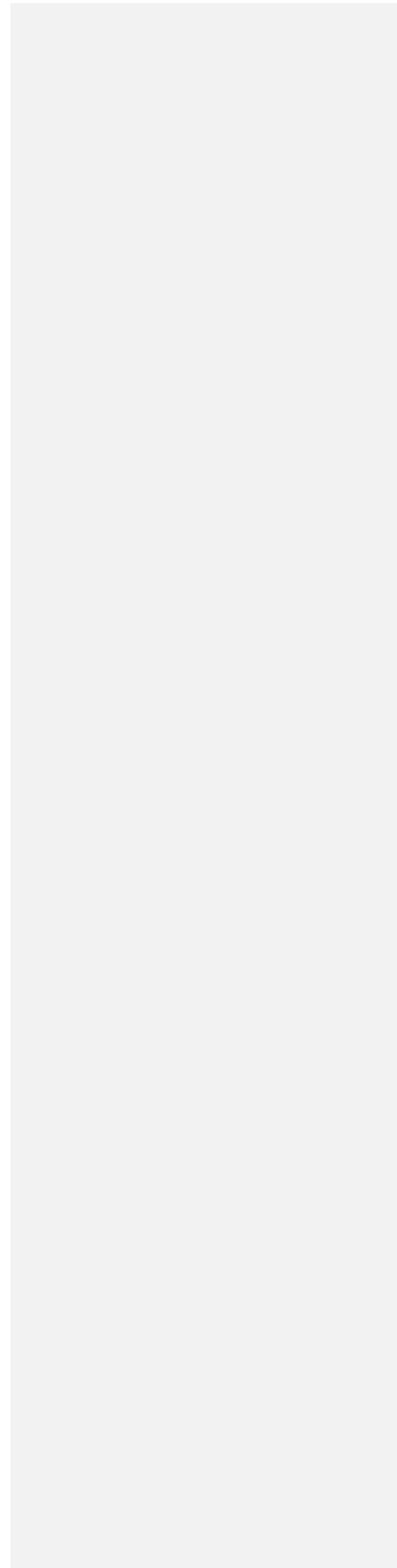


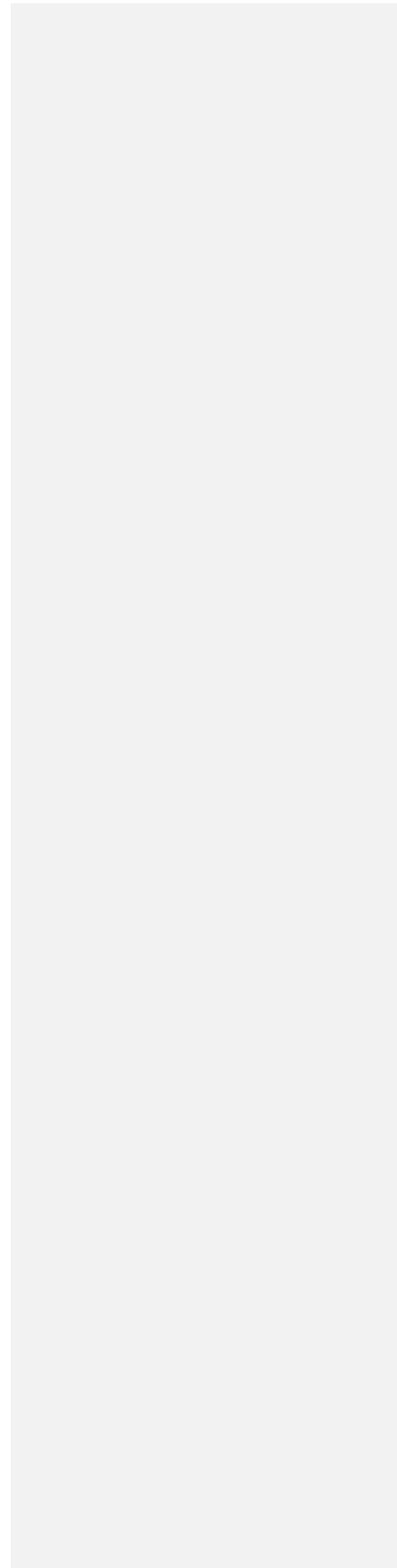
Referências

ALVES, E.M.; RANZI, T.A.D. (2006). Governança de TI: Avaliação de Maturidade do COBIT em uma empresa global, Universidade Federal de Santa Catarina.

Comment [F740]: Espaço grande

Comment [F741]: A fonte das referências ta diferente ou é assim mesmo ?





Page 104: [1] Comment [A511] **Alinne** **10/24/2009 1:50:00 PM**

Acho que ficaria melhor assim:

No entanto, ao tomar como referência a própria definição de Shepperd & Ince [Shepperd & Ince 1993], surge o questionamento sobre o que vem a ser medição e também, diretamente relacionado, o seu substantivo correlato: medida.

Page 104: [2] Comment [A512] **Alinne** **10/24/2009 1:56:00 PM**

Acho que ficaria melhor assim:

A partir dessa indagação, pode-se entender medição como o ato de obter valores de uma característica ou atributo de uma entidade qualquer, como por exemplo, uma pessoa. A respeito da entidade pessoa existe um conjunto de características ou atributos, os quais são passíveis de receberem valores, tal como: altura, peso, entre outros. Para estes atributos pode-se atribuir os valores 85 para o peso e 180 para altura. No entanto, para tais valores costumam-se atribuir unidades de medida, as quais são responsáveis por oferecer uma noção de quantidade de um valor qualquer e que servem de base para comparações com outros valores com mesma unidade de medida. Portanto, para este exemplo o peso seria igual a 85 kg e altura igual a 180 cm.

Page 104: [3] Comment [A513] **Alinne** **10/24/2009 2:01:00 PM**

O caso abordado anteriormente, também acontece com o software, onde o software e o processo usado para o desenvolvimento podem ser vistos como entidades, existindo atributos pertinentes para ambos que podem ser medidos. Segundo Pressman [Pressman 2006], o que já foi abordado pode ser exemplificado quando um único ponto de dados foi coletado (por exemplo, o número de erros descoberto em um único componente de software), uma medida foi estabelecida, e que uma métrica de software é aquela que relaciona medidas individuais de algum modo (por exemplo, o número médio de erros encontrados por teste de unidade).