Processos, Qualidade e Gestão de Software

v2 4 nov 2009

Indice

HERBSLEB, J. D., MOITRA, D. "GLOBAL SOFTWARE DEVELOPMENT", IEEE SOFTWARE, MARCH/APRIL, EUA, 2001, P. 16-20. **138**

KAROLAK, D. W. "GLOBAL SOFTWARE DEVELOPMENT – MANAGING VIRTUAL TEAMS AND ENVIRONMENTS". LOS ALAMITOS, IEEE COMPUTER SOCIETY, EUA, 1998, 159P. **138**

KIEL, L. "EXPERIENCES IN DISTRIBUTED DEVELOPMENT: A CASE STUDY", IN: WORKSHOP ON GLOBAL SOFTWARE DEVELOPMENT AT ICSE, OREGON, EUA, 2003, 4P. **138**

KIRCHER, M., JAIN, P., LEVINE, A. "DISTRIBUTED EXTREME PROGRAMMING", IEEE, AGILE 2008. **138**

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. **138**

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. **208**

- 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 ACHAR DEFEITOS O MAIS CEDO POSSÍVEL 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*. **209**

- 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 É TESTADA A INTEGRAÇÃO DOS COMPONENTES DO SOFTWARE ENTRE SI, MAS TAMBÉM DESTES 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. **209**

- 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 FAZ AQUILO QUE É SUPOSTO FAZER. 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

- TESTE DE ESTRESSE – TESTE CONDUZIDO PARA AVALIAR O COMPORTAMENTO DO SISTEMA DIANTE DE CONDIÇÕES QUE ULTRAPASSEM O LIMITE ESPECIFICADO NOS REQUISITOS. **210**

- 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. **210**

- 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. **210**

- 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 CORRETUDE 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. **210**

1.2.4. PROCESSO DE TESTES **210**

SEGUNDO [GRAHAM ET. AL 2007], 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*. ESTAS ATIVIDADES SÃO LOGICAMENTE SEQÜENCIAIS, PORÉM, EM UM PROJETO ESPECÍFICO, PODEM SE SOBREPOR, SEREM EXECUTADAS EM PARALELO OU ATÉ MESMO SEREM REPETIDAS. **210**

1.2.4.1. PLANEJAMENTO E CONTROLE **211**

DURANTE O PLANEJAMENTO DE TESTES DEVE-SE TER CERTEZA DE QUE OS OBJETIVOS DOS CLIENTES E *STAKEHOLDERS* FORAM ENTENDIDOS DE MANEIRA CORRETA. 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: **211**

- 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). 211

- 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. 211

- 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). 211

- 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. 211

- 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. 211

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. 211

1.2.4.2. ANÁLISE E PROJETO 211

ESTA É A ATIVIDADE EM QUE OS OBJETIVOS GERAIS DE TESTES SÃO TRANSFORMADOS EM CONDIÇÕES E PROJETOS DE TESTE TANGÍVEIS. 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: 211

- 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 IMPLEMENTAÇO, UMA VEZ QUE A BASE DE TESTES PODE SER USADA PARA COMPREENDER O QUE O SISTEMA PRECISA FAZER. 211

- IDENTIFICAR E DESCREVER CASOS DE TESTE: UM CASO DE TESTE É UM

IDENTIFICADOR, OBJETIVO, PRÉ-CONDIÇÕES DE EXECUÇÃO, ENTRADAS, PASSOS ESPECÍFICOS DO TESTE A SER EXECUTADO E RESULTADOS ESPERADOS E/OU PÓS-CONDIÇÕES DE EXECUÇÃO. UM CASO DE TESTE BEM PROJETADO TEM MUITA CHANCE DE ENCONTRAR UM ERRO AINDA NÃO CONHECIDO. **212**

- 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. **212**

- 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. **212**

1.2.4.3. IMPLEMENTAÇÃO E EXECUÇÃO **212**

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. 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. **212**

- IMPLEMENTAÇÃO: **212**

  - 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. **212**

  - 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. **212**

  - IMPLEMENTAR E VERIFICAR O AMBIENTE: PREPARAR E VERIFICAR SE O AMBIENTE DE TESTE ESTÁ FUNCIONANDO CORRETAMENTE. **212**

- EXECUÇÃO: **212**

1.2.4.5. ATIVIDADES DE ENCERRAMENTO DE TESTE 213

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. AS ATIVIDADES PRINCIPAIS SÃO: 213

- 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. 213

- 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. 213

- 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. 213

- 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. 213

1.2.5. TESTES AO LONGO DO CICLO DE VIDA DE SOFTWARE 213

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 CONTRARIO, NÃO CONSEGUIRÁ OBTER O SUCESSO DESEJADO. 213

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. A FIGURA 1.6 ILUSTRA O MODELO EM CASCATA. **214**

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. **215**

TODOS OS ARTEFATOS GERADOS PELOS DESENVOLVEDORES E ANALISTAS DE NEGÓCIO DURANTE O DESENVOLVIMENTO, PROVÊEM 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. **215**

1.3. INSPEÇÃO DE SOFTWARE **216**

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 TARDE, MAIS DIFÍCIL DE SE 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. **216**

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, [FAGAN 1986] A UTILIZAÇÃO DE INSPEÇÕES INFORMAIS DE SOFTWARE CAPTURAM EM TORNO DE 60% DOS ERROS EM UM PROGRAMA. MILLS ET AL. [MILLS ET AL. 1987] SUGERE QUE UMA APLICAÇÃO MAIS FORMAL DE INSPEÇÃO DE SOFTWARE PODE DETECTAR ATÉ MAIS DE 90% DOS ERROS DE UM PROGRAMA. SELBY E BASILI [SELBY ET AL. 1987] 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. **216**

1.3.1. A EQUIPE DE INSPEÇÃO (PARTICIPANTES) 217

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. UMA BOA VARIEDADE DE INSPETORES, REPRESENTANDO DIFERENTES ÁREAS DE CONHECIMENTO, É INTERESSANTE PARA O PROCESSO DE INSPEÇÃO. O PAPEL DE CADA PARTICIPANTE SERÁ EXPLICADO ABAIXO. 217

- 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. 217

- INSPETOR – EXAMINA O PRODUTO ANTES DA REUNIÃO DE INSPEÇÃO (FASE DE PREPARAÇÃO) E DURANTE DE MODO A TENTAR ENCONTRAR DEFEITOS. PODE TAMBÉM IDENTIFICAR PROBLEMAS AMPLOS QUE ESTÃO FORA DO ESCOPO DA EQUIPE DE INSPEÇÃO, COMO TAMBÉM SUGERIR MELHORIAS. 217

- 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. 217

- 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. 217

- 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. **217**

1.3.2. O PROCESSO DE INSPEÇÃO DE SOFTWARE (ETAPAS) 217

O PROCESSO TRADICIONAL DE INSPEÇÃO DE SOFTWARE [FAGAN 1976] É DEFINIDO POR SEIS ESTÁGIOS, CADA UM REPRESENTADO POR SEU PRINCIPAL RESPONSÁVEL. A FIGURA 1.8 ILUSTRA ESTA SEQÜÊNCIA DE ETAPAS E EM SEGUIDA CADA UMA DAS ETAPAS SERÁ EXPLICADA DETALHADAMENTE. **217**

- 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 CONTRARIO, O MATERIAL DEVERÁ SER DIVIDIDO EM TAMANHOS MENORES, COM INSPEÇÕES A SEREM REALIZADAS PARA CADA UMA DESTAS PARTES. **219**

- 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. **219**

- 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 CONTRARIO, ELE PODE REMARCAR A REUNIÃO. **219**

- 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. **219**

- 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. **219**

- 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, TODOS 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 UM TEMPO NA ETAPA DE RE-TRABALHO. **219**

1.3.3. FERRAMENTAS DE APOIO AO PROCESSO DE INSPEÇÃO 219

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: **220**

1. PRIMEIRAS FERRAMENTAS (*EARLY TOOLS*) **220**

2. FERRAMENTAS DISTRIBUÍDAS (*DISTRIBUTED TOOLS*) **220**

3. FERRAMENTAS ASSÍNCRONAS (ASYNCHROUNOUS TOOLS) **220**

4. FERRAMENTAS BASEADAS EM WEB (WEB-BSED TOOLS) **220**

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 FIM DA DÉCADA DE 90 SURGIRAM AS FERRAMENTAS PARA INTERNET. **220**

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. **220**

A SEGUIR SERÁ APRESENTADA UMA FERRAMENTA REPRESENTANTE DE CADA GERAÇÃO INTRODUZIDA ANTERIORMENTE. 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. **220**

- 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 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. **220**

- 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. **220**

- 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 INSPECÕES TAMBÉM ESTÃO PRESENTES NESTA

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** **234**

TÓPICOS DE PESQUISA **235**

O TMM (*TESTING MATURITY MODE*), COMO VISTO ANTERIORMENTE, FOI DESENVOLVIDO PELO *ILLINOIS INSTITUTY OF TECNOLOGY* COMO UM GUIA PARA MELHORIA DE PROCESSOS DE TESTES. EXISTE UMA ORGANIZAÇÃO 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. **235**

A FUNDAÇÃO TMMI TEM OS SEGUINTES OBJETIVOS: **235**

- DEFINIÇÃO DE UM NÚCLEO TMMI PARA A PADRONIZAÇÃO INTERNACIONAL **235**
- CRIAÇÃO E GESTÃO DE UMA ORGANIZAÇÃO INDEPENDENTE, IMPARCIAL COM REPOSITÓRIO CENTRAL DE DADOS E PRESTAÇÃO DE SERVIÇOS **235**
- MÉTODOS DE AVALIAÇÃO COM BASE NO MODELO PADRÃO **235**
- PRESTAÇÃO DE UM MECANISMO INDEPENDENTE PARA FACILITAR A VERIFICAÇÃO FORMAL DE AVALIAÇÕES TMMI **235**
- DEFINIÇÃO E MANUTENÇÃO DE AVALIADOR INDEPENDENTE **235**
- PRESTAÇÃO DE UM FÓRUM PÚBLICO DAS PARTES INTERESSADAS PARA FACILITAR A LIVRE TROCA DE INFORMAÇÃO, EDUCAÇÃO, IDÉIAS E USO DA NORMA PÚBLICA. **235**

O SITE OFICIAL DA FUNDAÇÃO É **HTTP://WWW.TMMIFOUNDATION.ORG**. **235**

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. **235**

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

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. **324**

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. **324**

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. **324**

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=S0103-6513200700 0200009&LNG=PT&NRM=ISO&TLNG=PT. ACESSADO EM: OUT. 2009. 324

CASTELO, L. GERÊNCIA PARTICIPATIVA: A COMUNICAÇÃO E O GERENTE. DISPONÍVEL EM: HTTP://WWW.GERANEGOCIO.COM.BR/HTML/GERAL/GP4.HTML. ACESSADO EM: SET. 2009. **325**

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. **325**

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. **326**

PMI (PROJECT MANAGEMENT INSTITUTE) A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE – GUIA PMBOK® 4. ED. UPPER DARBY, 2008. **326**

Parte 1

PROCESSOS

Capítulo
1

Processos tradicionais de desenvolvimento de software

Wislayne Aires Moreira

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.

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.

8. RUP

8.1 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.

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:

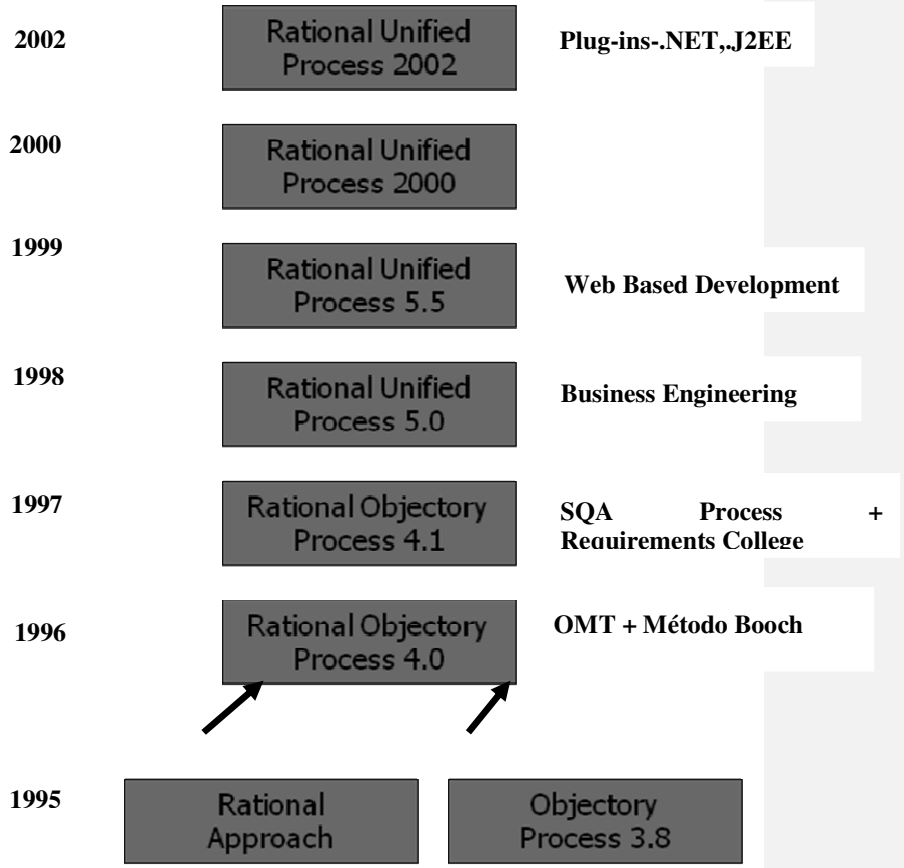


Figura 8.1. História do RUP

8.2 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 termino tem-se um produto

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.

8.3 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.

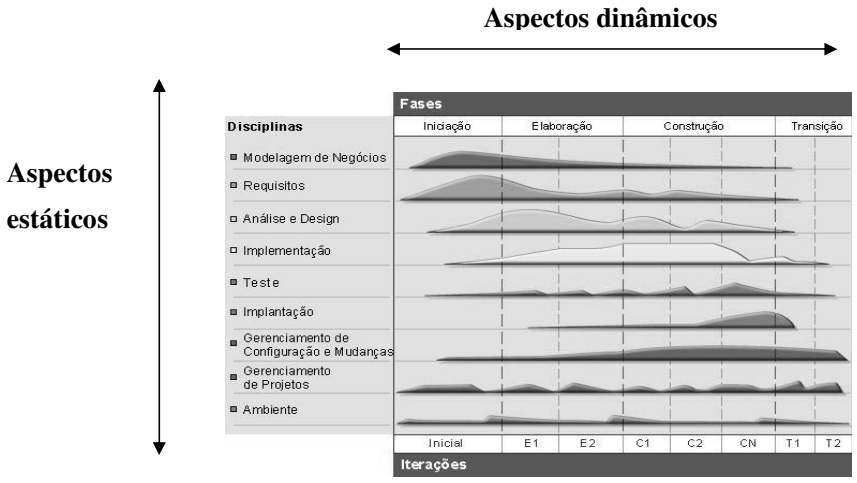


Figura 8.2 Arquitetura RUP

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:

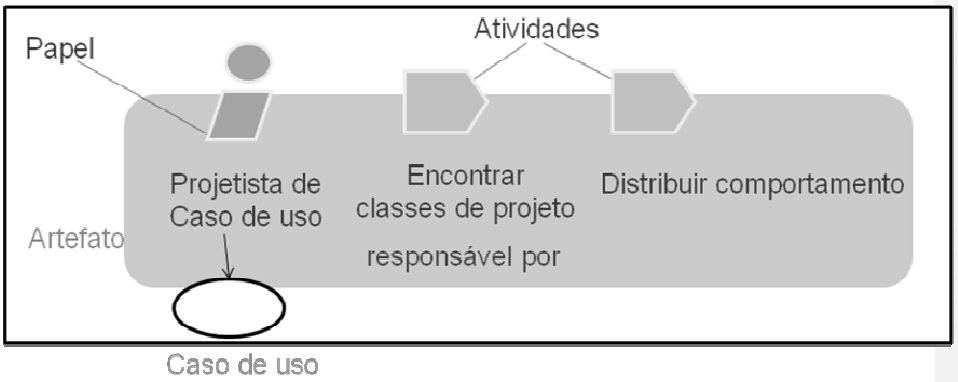


Figura 8.3 Descrição do processo

- 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 significante de atividades que produzem algum artefato e que mostram iterações entre os papéis.

8.4 Princípios básicos do RUP

Os seis princípios mais importantes do RUP:

Desenvolver software iterativamente,gerenciar requisitos,usar arquiteturas baseadas em componentes,modelar visualmente o software e verificação contínua da qualidade e controle de mudanças

8.2 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 e concebido pelo conceito de disciplina O real

seus objetivos da fase e ao mesmo tempo concluam 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.

8.2.1 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.

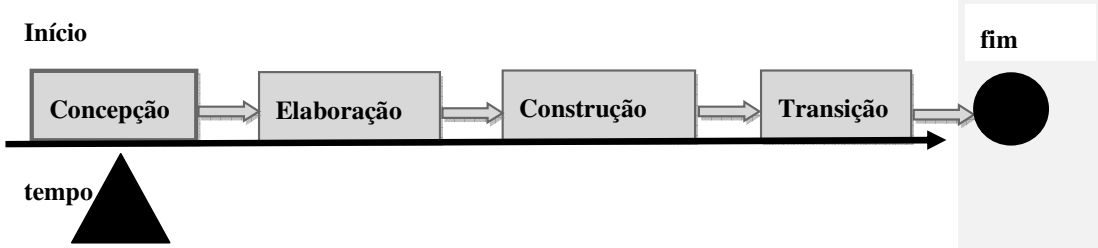


Figura 8.4 Fase de concepção

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 está 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.

8.2.2 Elaboração

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

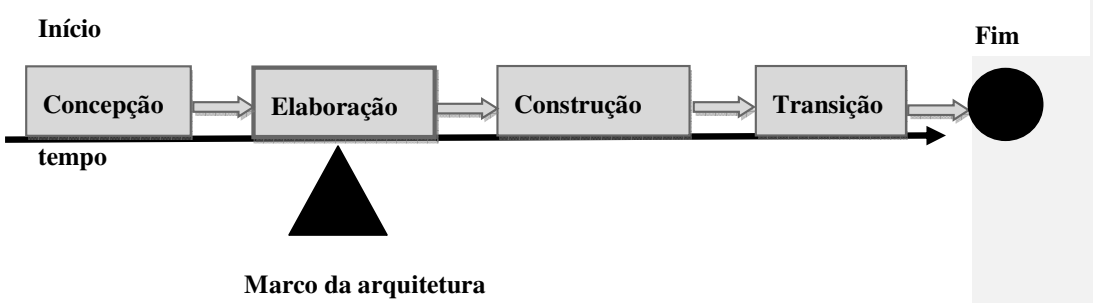


Figura 8.5 Fase de elaboração

O objetivo dessa 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 critica,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.

8.2.3 Construção

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

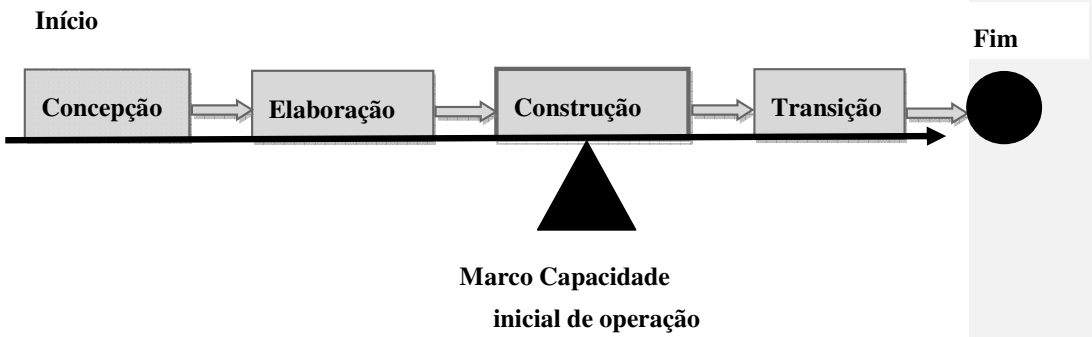


Figura 8.6 Fase de Construção

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

- As despesas reais com recursos ainda são aceitáveis se comparadas com as planejadas?

8.2.4 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.

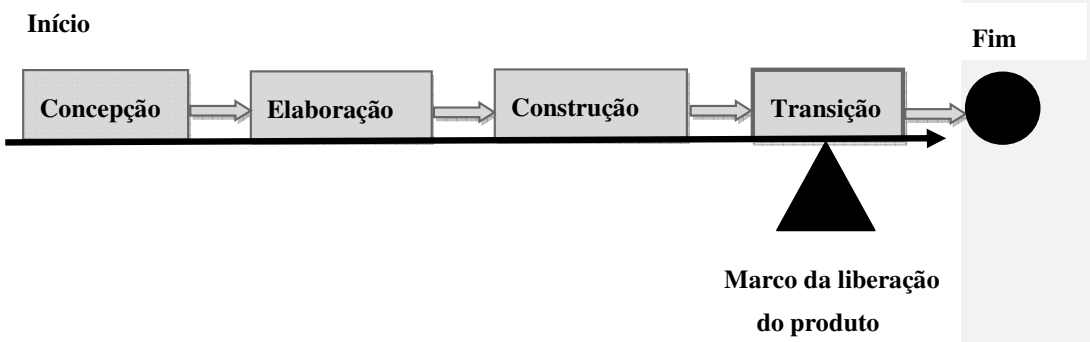
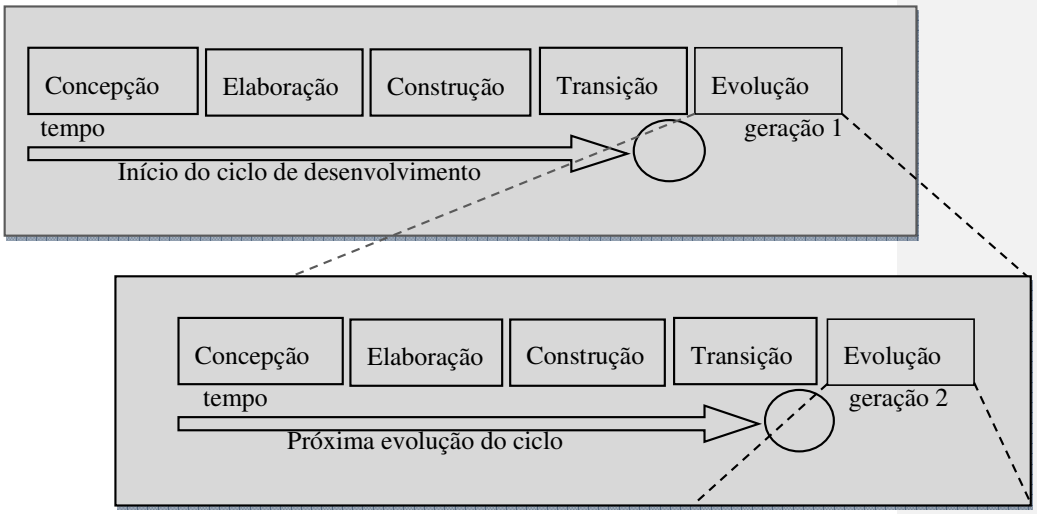


Figura 8.7 Fase de Transição

No final do ciclo de vida da fase de transição, os objetivos devem ter sido atendidos e o projeto deve está concluído. Em alguns casos o ciclo de vida atual pode combinar com o início de outro ciclo de vida,levando a 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.



A fase de transição pode ser mais simples ou mais complexa, depende do tipo do produto. Por exemplo, uma release de um produto de mesa, pode ser algo simples,enquanto que a substituição de um sistema de controle aéreo é algo complexo.

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.

8.3 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 6, 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 seqüencia 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

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 disciplina de implantação descreve 3 modos de implantação de produto: a instalação personalizada,o produto em uma forma "compacta" e acesso ao software por meio da internet.

  A ênfase 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.

9. O OpenUp

9.1 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 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.

9.2 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]

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.

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

9.3 Visão Geral do OpenUp

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

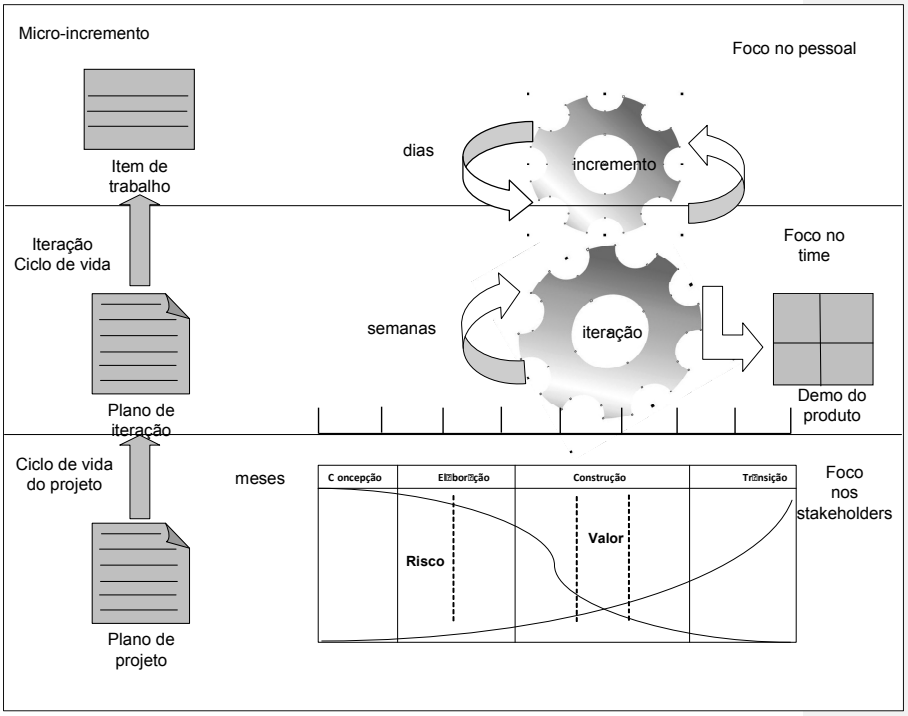


Figura 8.9 Visão Geral do OpenUp

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?

Ciclo de Vida da Iteração com foco no time:

Além da divisão por fases já conhecida, o OpenUP divide o projeto em iterações

podem variar de alguns dias a algumas semanas (a media 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.

9.4 As 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.

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.

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. Refine e detalhe o plano de projeto de alto nível.

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.

A fase de Transição possui alguns objetivos que ajudam a fazer um ajuste fino na funcionalidade,desempenho e na qualidade total do produto beta proveniente da fase anterior: **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 e **Melhorar o desempenho de projetos futuros com documentação das lições aprendidas** e melhorar o ambiente de processos e ferramentas para o projeto.

9.5 Princípios básicos do OpenUp

O OpenUP é baseado em 4 princípios básicos que são:

Equilíbrar 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 subseqüentes que definem a arquitetura do sistema.

De forma resumida podemos citar algumas práticas associadas a este princípio:

Conheça o ponto-alvo,analise custo-benefício,saiba quem são os *stakeholders* e o que eles realmente esperam do sistema.**Separe o problema da solução**,assegure de que tenha se entendido o problema antes de definir a solução,dessa forma nos ajuda a manter o foco apropriado e facilita acomodar as formas alternativas de resolver o problema.**Crie um entendimento compartilhado do domínio,**com esse entendimento conciso e compartilhado do domínio do problema eleva a comunicação e a efetividade do projeto.**Use casos de uso e cenários para capturar os requisitos** funcionais de

documentados nos Requisitos Suplementares,usando técnicas tradicionais.**Estabelecer e mantenha um acordo sobre prioridades** dos requisitos que serão implementados trabalhando regularmente com os *stakeholders* durante a evolução do produto.**Faça análises para maximizar o valor,**análise essa de custo-benefício não pode ser realizada independentemente da arquitetura. Os requisitos estabelecem os benefícios do sistema para o *stakeholder*, enquanto a arquitetura estabelece o custo. O custo de um benefício pode influenciar no valor percebido do benefício pelo *stakeholder*.**Gerenciamento de mudanças** apresentam oportunidades para elevar o valor para os *stakeholders*,mudanças irrestritas resultarão num sistema inchado,deficiente e inadequado para atender às necessidades do *stakeholder* e **saiba quando parar** o desenvolvimento de sistema,para evitar desperdícios de recursos e complexidade do sistema.

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.

Algumas práticas que aplicadas a este princípio podem ser citadas:

9.6 Disciplinas

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

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.

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, re-planejando 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.

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.

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 à seqüência 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 valoroso e que os membros da equipe de projeto entendam as necessidades dos *Stakeholders,*produzindo continuamente um produto de software para avaliação.

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.

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.

10. MSF

10.1 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 especificas 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 muito 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

MSF se misturam, podemos então considerar o MSF como a “atitude mental da Microsoft” para seu modelo de negócios.

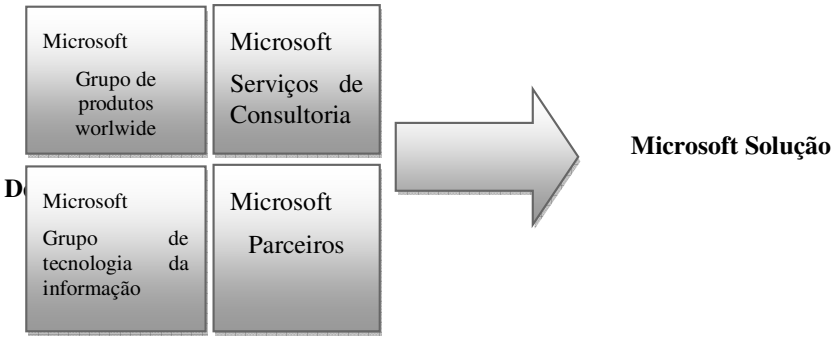


Figura 8.10 Microsoft Solução de framework

10.2 MSF e suas características

O MSF é um processo iterativo e incremental,onde as atividades são realizadas de forma cíclica e resultando em produtos incrementai,.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;o aprendizado contínuo,onde o projeto em uma grande lição a cada iteração. Tem-se a percepção da evolução da curva de produtividade e pods-se acompanhar a autonomia técnica crescente dos colaboradores a cada dia dentro do projeto e uma teoria de controle,onde pequenas iterações permitem calibrar com mais precisão e agilidade estimativas e reduzir margens de erro.

10.3 Visão Geral do MSF

Um projeto MSF é regido por iterações ou ciclos. A 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.

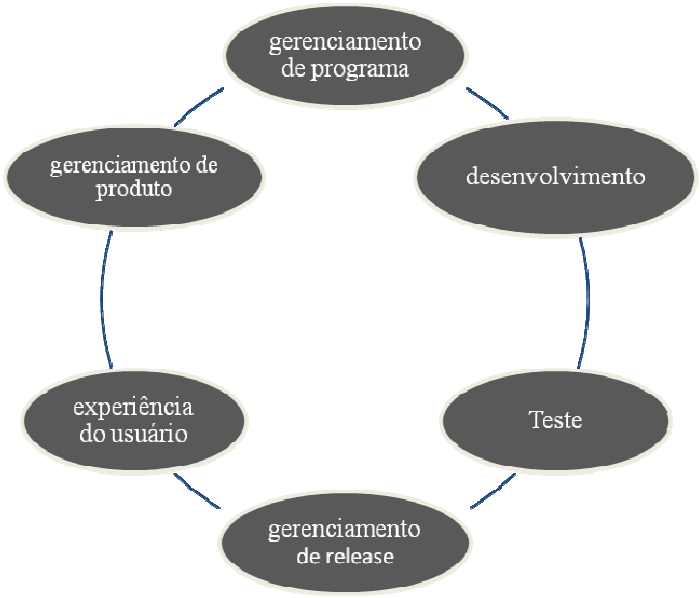


Fig 8.11. Modelo de times

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.

10.4 As fases do MSF

8.11. Fases do MSF

O modelo de processos do MSF em 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:

10.4.1 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 projeto deve fazer) e a definição do escopo do projeto.

10.4.2 Planejamento: Durante essa fase a equipe estima custos,prepara plano de trabalho e programa os deliverables.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.

10.4.3 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.

10.4.4 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.

10.4.5 Implantação:nessa fase, a equipe estabiliza o produto e obtém a aprovação do

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 coletada em todas as fases.

10.5 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.

10.5.1 Parceria com o cliente

A idéia 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.

10.5.2 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.

10.5.3 Trabalho em direção a uma visão compartilhada

Segundo Steve McConnel,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.

10.5.4 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 mitigar os impactos negativos.

10.5.5 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 minimizar o desconhecimento,incertezas e retrabalho.

10.5.6 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

10.5.7 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 principio diminui as incertezas quanto "o que", "quem", "quando" e "por que" com os resultados,tornando o trabalho mais eficiente e compensador.

10.5.8 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 intima com o negócio,se não existir essa ligação pode resultar em entregas com atraso e projetos cancelados.

10.5.9 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.

10.5.10 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.

10.6 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:

10.6.1 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 individuo 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.

10.6.2 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.

10.6.3 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.

10.7 Exercícios

1. Por que o RUP é considerado um processo tradicional? Justifique a sua afirmação.

_____
_____
_____
_____
_____
_____

2. O que faz do processo RUP um modelo iterativo e incremental?

_____
_____
_____
_____

3. As fases do RUP são definidas em 4.Quais são os seus objetivos 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. O que faz do OpenUp um processo tradicional ? Justifique sua resposta.

7.Fale sobre os princípios básicos do OpenUp.

_____
_____
_____
_____
_____

8. Quais são as fases do MSF? Detalhe cada uma.

_____
_____
_____
_____
_____

9. Fale um pouco sobre o modelo de time do MSF.

_____
_____
_____
_____
_____
_____

10. Explique os princípios do MSF. Selecione os que você acha mais importantes.

_____
_____
_____
_____

10.8 Sugestão de Leitura

Introdução ao RUP:Rational Unified Process-“ Phillippe Kruchten”

Conheça o Rational Unified Process-“Mauro Viana” 2004

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

10.9 Tópicos de Pesquisa

1. http://groupware.les.inf.puc-rio.br/groupware/publicacoes/2008.SBSI.Pimentel.RUP3CGroupware.pdf

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

3. http://www.cingo.com.br/

10.10 Referências Bibliográficas

http://www.wthreex.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

http://www.laps.ufpa.br/yomara/paginay2/aps/processo%20unificado%20rup

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://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%C3%81geis%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

Índice do Capítulo

Comment [wis1]: Formatar o índice

Capítulo

3

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. Neste capítulo são citadas as principais características das Metodologias Ágeis em geral, com ênfase na Extreme Programming (XP), Scrum e Feature Driven Development (FDD).

3.1. Introdução a Processos Ágeis de Desenvolvimento de Software

Os processos tradicionais de desenvolvimento de software geralmente não se adéquam 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 flexível à mudança e privilegiando as atividades que agregam valor ao negócio.

Tanto os métodos pesados, tanto 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.

Comment [wis2]: Acho que não cabe aqui.Melhor mudar para Quanto.

Comment [wis3]: Ficou Vago.

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

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.

Comment [wis4]: Começar no mesmo parágrafo

As metodologias ágeis surgiram com a finalidade de desburocratizar o processo de desenvolvimento. Elas tentam se adaptar e fortalecer com as mudanças, até mesmo ao 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.

Outro fator determinante da agilidade 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.

Comment [wis5]: Fala no mesmo assunto

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.

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

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar [AGILE MANIFESTO 2009]:

Comment [wis6]: Texto incompleto.

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,

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:

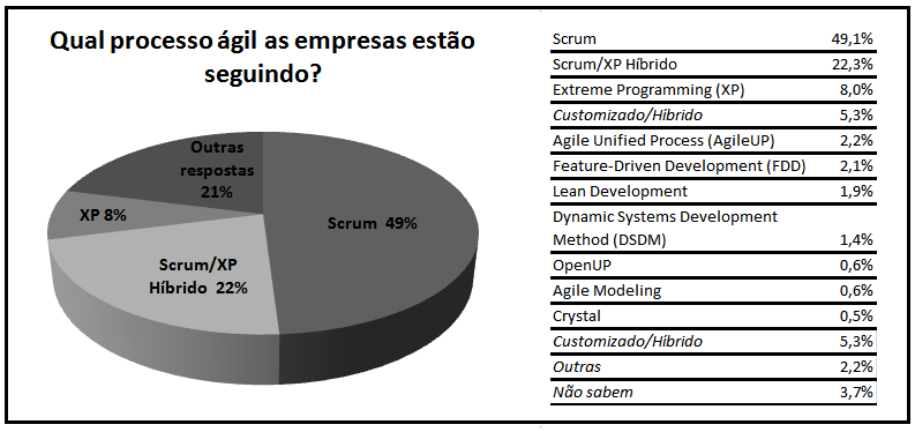
- Nossa maior prioridade é satisfazer o cliente através de entregas antecipadas e contínuas de software de valor ao cliente;
- Mudanças de requisitos são bem vindas, mesmo que tardiamente no desenvolvimento. Processos ágeis se aproveitam da mudança para vantagem competitiva do cliente;
- Entrega freqüente 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;
- 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.

3.3. Principais Processos Ágeis

As metodologias que seguem os princípios do Manifesto Ágil – como entrega freqüente, flexão a mudança, foco em pessoas e simplicidade – são consideradas Processos Ágeis de Desenvolvimento de Software.

Comment [wis7]: Falta complement.

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.



Scrum	49,1%
Scrum/XP Híbrido	22,3%
Extreme Programming (XP)	8,0%
Customizado/Híbrido	5,3%
Agile Unified Process (AgileUP)	2,2%
Feature-Driven Development (FDD)	2,1%
Lean Development	1,9%
Dynamic Systems Development Method (DSDM)	1,4%
OpenUP	0,6%
Agile Modeling	0,6%
Crystal	0,5%
Customizado/Híbrido	5,3%
Outras	2,2%
Não sabem	3,7%

Figura 3.1. Metodologias Ágeis mais utilizadas nas empresas de acordo com a 3° 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 IBM Rational Unified Process (RUP), que estar 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, mes mo 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. 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 [wis8]: Acho que não coube aqui.Faltou colocar uma frase antes para preparar o leitor sobre o que iria falar.

O Crystal não é apenas uma única metodologia, e sim, uma família de métodos denominada portanto de Família Crystal, proposto por Alistair Cockburn. É organizada por cores de acordo com o número de pessoas envolvidas, o que se traduz em diferentes níveis de ênfase na comunicação. A versão mais ágil e mais documentada é a Crystal Clear que pode ser usada em projetos de até 8 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), etc.

Comment [wis9]:

Nas próximas seções deste capítulo serão tratadas com maior detalhamento as

grandes projetos e para empresas que possuem dificuldades para migrar para um ambiente completamente ágil.

3.4 EXTREME PROGRAMMING (XP)

Extreme Programming é considerada uma metodologia leve de desenvolvimento de software, esta metodologia é chamada de leve por que sempre que há mudanças o projeto está leve de documentação para ser modificada. Esta metodologia nasceu com Kent Beck e Ward Cunningham a partir da observação das necessidades decorrentes do surgimento de uma outra opção além daquelas oferecidas pela engenharia de software corrente que não traziam bons resultados.

O termo "Extreme" indica que a metodologia emprega ao extremo as boas práticas de engenharia de software. Esta metodologia tem algumas práticas mais peculiares do que aquelas observadas na metodologia ágil como programação em pares, forte cultura de testes que são automatizados e executados várias vezes ao dia e propriedade de código coletivo, detalharemos esta metodologia a seguir.

3.4.1 Valores, princípios e práticas de XP

Extreme programming é definida através de valores, princípios e práticas. Os valores descrevem os objetivos de longo prazo de quem aplica XP e definem critérios para se obter sucesso. Os quatro valores de XP são:

- **Comunicação –** Parte do insucesso de alguns projetos de software é atribuído a falta de comunicação. Por isto, a metodologia emprega algumas práticas que forçam uma maior comunicação por parte da equipe;
- **Simplicidade –** O mais simples possível que seja funcional. Para isto, deve-se desenvolver pensando apenas no presente. Pois nem sempre o software será modificado;
- **Retorno –** Assim como o valor comunicação, XP estabelece várias práticas para que o retorno freqüente. Retorno do Cliente, andamento do projeto, resultados dos testes tudo é devidamente medido e repassado a toda equipe para que haja uma melhor resposta;
- **Coragem –** A probabilidade de um código ser jogado fora por uma insatisfação do cliente, ou por mudanças de requisitos é muito grande, também é possível ter várias opções de implementação e realizar um pouco de cada uma para escolher a melhor.

Para sustentar estes valores a metodologia define princípios que devem ser seguidos por todos os praticantes:

- **Retorno rápido** – Além de obter retorno este deve ser rápido;
- **Assumir simplicidade** – Os problemas devem ser tratados da forma mais simples possível. XP prega que o tempo gasto pensando em reuso ou resolvendo possíveis problemas futuros pode ser maior do que ter o retrabalho de fazer *refactoring*;
- **Mudança incremental** – Mudanças devem ser feitas pouco a pouco, ou seja, de forma incremental e contínua;

Comment [wis10]: Onde estão as pontuações?

Comment [wis11]: Poderia está no mesmo parágrafo.

Comment [wis12]: Pelo que percebi o pois é uma conjunção coordenada que pode ser conclusiva ou explicativa. Serve como conector entre as duas orações.

Comment [wis13]: Falta complemento.

Comment [wis14]: Acho que está muito sintetizado.

- **Trabalho de qualidade** – Deve-se sempre buscar produzir um trabalho de qualidade, de outra forma a equipe perderá a motivação necessária ao projeto;
- **Ensinar aprendendo** – XP faz com que o praticante aprenda suas próprias medidas daquilo que deve projetar, do quanto se deve testar e de como fazer *refactoring*;
- **Investimento inicialmente pequeno** – Um projeto deve iniciar com poucos recursos e no decorrer, com sucesso do projeto e retorno do cliente, ir aumentando estes recursos. Um projeto pode ser cancelado ou ter seu escopo diminuído, nestes casos é melhor encerrá-lo com um investimento ainda pequeno;
- **Jogar para vencer** – A equipe deve ser vencedora, ou seja obter sucesso ao fim do projeto;
- **Experimentos concretos** – Decisões abstratas devem ser testadas em forma de experimentos;
- **Comunicação aberta e honesta** – Problemas como atrasos, má qualidade de código e insegurança devem ser tratados abertamente pelo grupo de projeto;
- **Trabalhar a favor dos instintos das pessoas** – Visto que os instintos são baseados em seus interesses de curto prazo, como vontade de vencer e aprender a interagir com outras pessoas, deve-se respeitá-lo e utilizá-lo a seu favor;
- **Aceitar responsabilidades** – Responsabilidades devem ser aceitas e não impostas. Isto leva as pessoas a um maior comprometimento com o projeto;
- **Adaptação ao local** – XP deve ser adaptada ao local de trabalho em que será empregada;
- **Viajar leve** – A equipe deve gerar poucos e valiosos artefatos. Como os projetos de XP têm que se adaptar às várias mudanças necessárias em um projeto, a equipe deve estar "leve" de artefatos, documentações e especificações. De outra forma será empregado muito esforço para atualizá-los;
- **Medidas honestas** – As métricas a serem utilizadas devem ser o mais honestas possíveis e refletir as necessidades do projeto.

Comment [wis15]: Acho que deveria mudar o nome da prática.

Extreme Programming define também práticas que tornam a metodologia viável e possível de seguir seus valores e princípios. Abaixo estão listadas todas as práticas que são:

- **O jogo do planejamento** - Deve ser elaborado de forma simples, fácil e rápido o plano para o próximo *release*, que consiste em determinar o escopo baseado nas prioridades do negócio, nas possibilidades e estimativas técnicas;
- ***Releases* pequenos** – Os *releases* devem ser de pouca duração e devem sempre conter software de valor;
- **Metáforas** – Guiam o desenvolvimento do projeto através de uma estória que explique como o sistema funciona;
- **Projeto de software simples** – O sistema deve ser projetado da forma mais simples que possa solucionar o software.
- **Teste** – Há dois estágios de testes definidos: testes unitários e testes de aceitação. Ambos devem ser automatizados. Os testes unitários são feitos pelo programador durante o desenvolvimento. Os testes de aceitação são especificados pelo cliente ditando que deve funcionar para que o software seja

- ***Refactoring*** – Reestruturação de parte do código, sem alterar seu comportamento;
- **Programação em pares** – Todo o código é produzido por pares de programadores, cada par trabalhando na mesma máquina;
- **Propriedade coletiva** – Todos os integrantes da equipe são donos e responsáveis pelo código produzido. Assim, todos podem alterar qualquer parte do código a qualquer momento;
- **Integração contínua** – O código é integrado, o *build* do software é gerado e os testes são executados várias vezes ao dia;
- **40 horas semanais** – Os membros da equipe não devem trabalhar mais que quarenta horas semanais;
- **Cliente no local –** O cliente é parte integrante da equipe e deverá estar presente no local de desenvolvimento sempre que necessário para tirar alguma dúvida ou priorizar alguma estória;
- **Padrão de codificação** – O código é utilizado como fonte de informação para comunicação e discussão entre os membros da equipe. Assim, ele deve estar bem escrito e estruturado, o que requer a conformidade a um padrão de codificação estabelecido para todo projeto.

Comment [wis16]: Colocar em português.

3.4.2 Papéis dos integrantes

XP relaciona papéis, com responsabilidades explícitas, a serem desempenhados em um projeto. Os mais importantes são:

Comment [wis17]: Precisa de mais um incentive ao leitor. Colocar um texto antes.

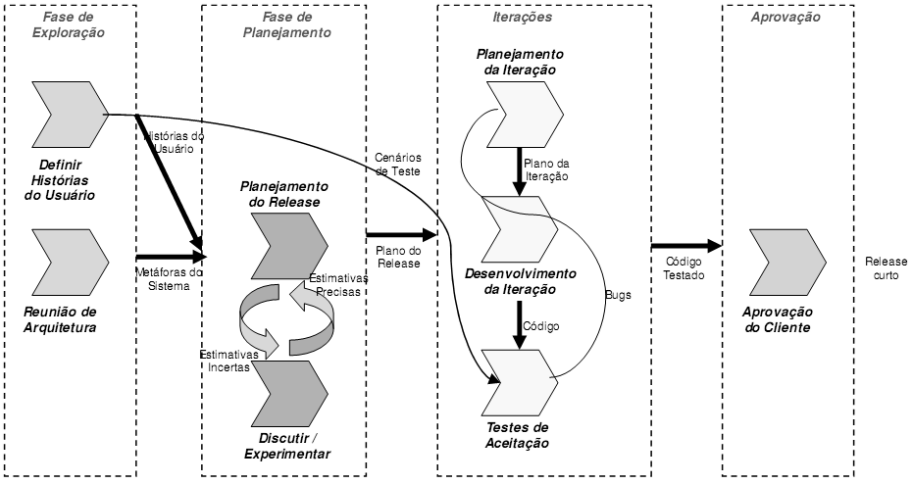
- **Programador** – É o coração de XP. Responsável por projetar o software, codificar, implementar testes e estimar suas tarefas. Todos da equipe assumem este papel no desenvolvimento do projeto;
- **Cliente** – Representante do cliente, responsável por estabelecer prioridades e escopo, escrever estórias, escrever os testes funcionais. Deve estar disponível no local do desenvolvimento sempre que necessário para esclarecer duvidas;
- **Testador –** O testador é responsável por apoiar o cliente a escolher e escrever os testes funcionais, além de assegurar a execução e reportagem dos problemas identificados nestes testes;
- **Rastreador (*tracker*) –** É a consciência da equipe XP. Responsável por reportar as métricas do projeto promovendo visibilidade sobre a acurácia das estimativas e progresso do projeto;
- **Técnico –** Praticamente um gerente de projeto. Responsável por identificar problemas e resolvê-los para que a equipe possa trabalhar da melhor forma. Não requer conhecimentos técnicos profundos, mas, não se deve negar que conhecimento técnico ajuda no papel.
- **Consultor –** Papel que é opcional em uma equipe em XP. Ele age ajudando os outros a resolverem seus próprios problemas. O Consultor é um papel de profundo conhecimento técnico que aparece quando um membro da equipe tem um problema a resolver. Ele ajuda o membro a resolver o problema, joga a

- **Chefe –** O Big Boss tem um papel de gerente sênior, é ele quem toma decisões de mais alto nível e se reporta ao cliente, esse papel não exige conhecimento apesar de se o chefe o tiver é muito valioso. O Chefe precisa de coragem, confiança no time e muita insistência.

Existe também a possibilidade, caso seja desejo da equipe, que os membros possam "trocar de boné" dentro da organização. Ou seja, pessoas diferentes troquem de papel dentro da mesma organização.

3.4.3 Ciclo de Vida

Comment [wis18]: Falta Escrever o ciclo de vida



3.6. 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 a identificação e correção de quaisquer deficiências e/ou impedimentos no processo de desenvolvimento. [SCHWABER 2008]

Scrum vem sendo largamente utilizando em organizações ao redor do mundo. Ele permite manter o foco na entrega do maior valor de negócio, no menor tempo possível

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.

3.6.1 Características do Scrum

Trata-se de uma abordagem empírica focada nas pessoas para ambientes em que os requisitos surgem e mudam rapidamente, resultando em uma abordagem que reintroduz as idéias de flexibilidade, adaptabilidade e produtividade [BEEDLE e SCHAWABER 2002].

Comment [wis19]: Pontuação.

Segundo [BEEDLE e SCHAWABER 2002], a metodologia baseia-se em princípios como: equipes pequenas de, no máximo, 10 pessoas, requisitos que são pouco estáveis ou desconhecidos e iterações curtas. Divide o desenvolvimento em intervalos de tempos de, no máximo 30 dias, também chamadas de Sprints. Esta metodologia não requer ou fornece qualquer técnica ou método específico para a fase de desenvolvimento de software, apenas estabelece conjuntos de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto.

Comment [wis20]: Colocar no mesmo parágrafo.

3.6.2 Papéis do Scrum

O Scrum define para sua estrutura iterativa e incremental três papéis principais para diferentes tarefas, propósitos do processo e suas práticas: Scrum Master, Product Owner, Team (SCHWABER 2008).

Comment [wis21]: Colocar em português.

• **Scrum Master:** Gerencia o processo, ensinando o Scrum a todos os envolvidos no projeto e implementando o Scrum de modo que esteja adequado à cultura da organização; deve garantir que todos sigam as regras e práticas do Scrum; é responsável por remover os impedimentos do projeto (SCHWABER 2008). Ele é o líder e facilitador para o Team e Product Owner, responsável por (BIRD e SCHAWABER 2008):

1. Resolver barreiras entre o Team e o Product Owner;

2. Ensinar o cliente a maximizar o retorno sobre o investimento (ROI);

3. Garantir que o processo seja seguido;

4. Melhorar a vida da equipe de desenvolvimento, facilitando a criatividade e a capacitação;

5. Melhorar a produtividade da equipe de desenvolvimento de qualquer forma possível;

6. Melhorar as práticas de engenharia e prover ferramentas de modo que cada nova funcionalidade seja potencialmente realizada;

7. Manter as informações sobre os progressos da equipe até a data atual e proporcionar esta visibilidade a todas as partes envolvidas no projeto.

• **Product Owner:** É o responsável pelo retorno sobre o investimento (ROI), ou seja, seu foco é na parte comercial do produto. Ele é o representante de todos os stakeholders

estar à disposição da equipe em qualquer momento, mas, sobretudo durante o Sprint Planning Meeting e Sprint Review Meeting (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.

• **Team:** É um grupo de pessoas com diferentes habilidades necessárias para transformar requisitos em um incremento potencialmente entregável. Para tanto, geralmente são uma mescla de analistas, designer, gerente de qualidade, desenvolvedor etc. A equipe tem a autoridade de decidir, quando necessário, as ações que serão realizadas e priorizá-las organizando-as em Sprints. Effort estimation, Sprint Backlog, revisões de product Backlog List e sugestões de impedimentos para serem removidos do projeto também são atividades do time (SZALVAY 2007).

3.6.3 Práticas do Scrum

Comment [wis22]: Falta escrever

3.6.4 Ciclo de Vida do Scrum

Comment [wis23]: Falta escreever

3.7 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 o *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.

Comment [wis24]: Colocar no mesmo parágrafo.

Stephen Palmer [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.

3.7.1 Características do FDD

As metodologias ágeis devem seguir o todo ou parte do que foi acordado no Manifesto Ágil, seção introdutória desse capítulo, mesmo surgidas antes do referido manifesto e por isso tendem a possuir características comuns [BECK, FOWLER 2001].

Comment [wis26]: Acho que precisa de mais texto.

Algumas características intrínsecas nos processos ágeis são levantadas por Pekka Abrahamsson [2003], a saber:

- Cliente presente;
- Iterações curtas;
- Equipes pequenas (<12 aproximadamente);
- Entregas frequentes do produto;
- Adaptativos as mudanças;
- Flexibilidade; e
- Simplicidade.

Porém Craig Larman [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; e
- 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.

O FDD apresenta em seu escopo a definição de papeis para que se possa ter uma maior organização e visão na hora de se pensar/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 features 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 features 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

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

3.5.3 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 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: Qualquer atividade a ser desenvolvida deve ser analisada para verificar se esta pode ser quebrada em atividades menores, ou funcionalidades atômicas (indivisíveis). Essa prática torna o código inteiro mais seguro contra erros. Além do mais, garante flexibilidade e escalabilidade ao código;

- Posse individual do código: os desenvolvedores possuem individualmente posse de código. O proprietário do código torna-se automaticamente responsável pela performance, consistência, corretude e integração da classe. O proprietário é aconselhado a utilizar os melhores padrões da engenharia de software;

- Equipes de funcionalidades: são equipes pequenas responsáveis por desenvolver uma pequena atividade. Dessa forma, a equipe decide de forma conjunta como será feito o design de determinada funcionalidade. Com mais de uma pessoa pensando no mesmo problema, soluções diferentes podem surgir. O que se torna muito eficiente, visto que no final essas soluções podem ser fundidas em uma solução final da equipe;

- Inspeções: inspeção de código é uma ótima prática de engenharia, pois além de fazer verificações contra erros, incentiva uma melhor modelagem do sistema, junto com atributos como legibilidade e alta coesão.

- Gerência de configuração: essa prática tem como objetivo manter um controle sobre o código fonte das funcionalidades implementadas, mapeando as funcionalidades aos seus respectivos códigos-fontes e aos seus proprietários. Além disso, mantém as datas de modificação do código, guardando um histórico de alterações;

- Build constante: deve existir sempre uma versão do sistema rodando numa máquina. Isso garante que a equipe possui pelo menos uma versão do sistema que funciona. Dessa forma, sempre que for necessário apresentar alguma funcionalidade para o cliente, existirá uma versão do sistema que pode ser utilizada para isso; e

- Visibilidade do progresso: Como sempre é mantido um relatório de progresso das atividades do projeto, todos na equipe e fora dela sabem exatamente como estão em termos de produtividade. No entanto, essa atividade deve ser feita com muita precisão e frequência. Caso contrário, os dados extraídos do relatório serão enganosos e poderão levar a decisões desastrosas para o cliente.

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

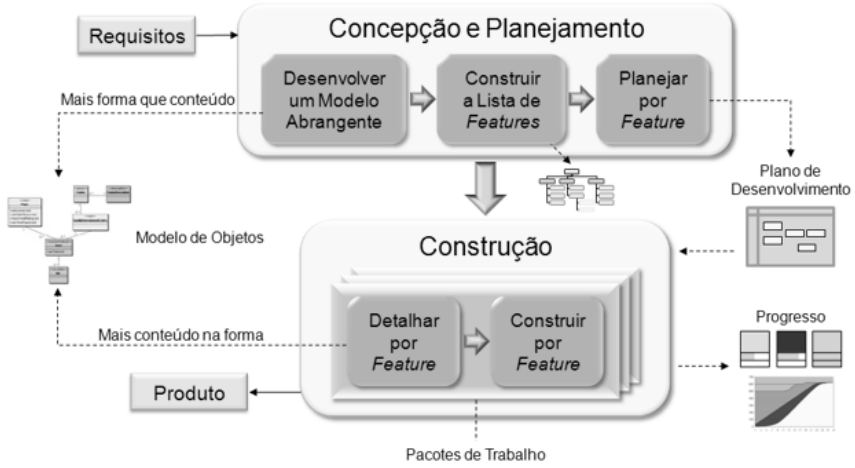
3.5.4 Ciclo de Vida do FDD

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

Comment [wis27]: Em baixo tem modelo geral.

Comment [wis28]: Colocar o mesmo nome nos texto abiaxo.

Comment [wis29]: Coloca r embaixo da figura(Fig X….)



Desenvolver um modelo geral: o projeto começa com uma análise superficial do escopo do sistema e seu contexto. Assim, são estudados os domínios de negócio do sistema e criado um modelo geral baseado nestes. Depois é criada uma modelagem superficial para cada área de domínio existente. Cada um desses modelos é revisado por um grupo aleatório de membros do projeto e melhorias são discutidas. É escolhido

domínio são fundidos para gerar um modelo do domínio como um todo (ou domínio principal) do sistema.

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: o conhecimento obtido na fase de desenvolvimento do modelo geral é essencial 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 gere valor para o cliente. Devem seguir o formato <ação> <resultado> <objeto>, por exemplo: "Gerar relatório de vendas" ou "Validar a senha do usuário". Essas funcionalidades são muito importantes para o processo como um todo. A não identificação delas causa um impacto enorme sobre projeto, pois significa que a primeira fase não foi feita com sucesso e consequentemente o efeitos aparecem em cascata nas próximas fases.

Sub-atividades:

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

Planejar por funcionalidade: é realizado o planejamento de desenvolvimento de cada funcionalidade da lista obtida da fase anterior. São designadas classes ou código específico para os programadores-chefe tomarem conta. Daí em diante, os programadores-chefe serão responsáveis pelo código produzido nessas classes.

Sub-atividades:

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

Modelar por funcionalidade: os programadores-chefe escolhem algumas funcionalidades para que, junto com os proprietários de código, sejam feitos os

funcionalidade em questão. Ou seja, nesta etapa pensa-se nas classes, métodos e atributos que irão existir. Ao final da modelagem, é realizada uma inspeção do modelo pela equipe que a fez ou por outra equipe designada.

Sub-atividades:

- 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: após a modelagem, o código é finalmente implementado e os testes finalmente escritos. Assim, o código fonte é produzido e as funcionalidades ganham vida. O programador-chefe designa um programador para implementar uma certa funcionalidade. Logo, este último será o proprietário do código escrito. Após uma inspeção no código escrito, a funcionalidade é terminada.

Sub-atividades:

- Implementar as regras de negócio das classes;
- Inspecionar código;
- Conduzir testes unitários;
- 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.

9.7. Considerações Finais

<PENDENTE>

9.8. Tópicos de Pesquisa

<PENDENTE>

9.9. Sugestões de Leitura

<PENDENTE>

9.11. Exercícios

<PENDENTE>

Referências

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.

AGILE MANIFESTO (2001). http://www.agilemanifesto.org/ - Último acesso em 18/10/2009.

VERSIONONE (2008). 3º Annual Survey: The State Of Agile Development. Disponível em: http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf

AMBLER, S. (2002) Agile Modeling. New York: Wiley Computer Publishing.

SCHWABER, K. (2008) *Agile Project Management With Scrum.* Redmond: Microsoft Press.

SCHWABER, K., BEEDLE, M. (2001) Agile Software Development with Scrum. City: Prentice Hall.

KOSCIANSKI, A., SOARES, M. (2006) Qualidade de Software. 2. ed. São Paulo: Novatec.

ABRAHAMSSON, P., WARSTA, J., SIPONEN, M.T., & RONKAINEN, J. New Directions on Agile Methods: A Comparative Analysis. In: ICSE 2003, USA

BECK, Kent; FOWLER, Martin. Planning Extreme Programming. 1. ed. Boston: Addison-Wesley, 2001.

COAD, Peter; LEFEBVRE, Eric; LUCA, Jeff. Java Modeling In Color With UML: Enterprise Components and Process. Upper Saddle River, N.J.: Prentice Hall, 1999.

FOWLER, M.. The New Methodology. Disponível em: http://martinfowler.com/articles/newMethodology.html. Acesso em: 15 Set 2009.

HIGHSMITH, J. Agile software development ecosystems. Boston, MA., Pearson Education, 2002.

LARMAN, Craig. Agile and iterative development : a manager's guide. Addison-Wesley, 2003.

PALMER S. R., FELSING J. M. A Practical Guide to Feature-Driven Development (The Coad Series) , Prentice Hall PTR, USA, 2002.

SLIGER, Michele; BRODERICK, Stacia. The Software Project Manager's Bridge to Agility. Addison Wesley Professional, 2008.

Comment [wis30]: Formatar a referência

Índice

Capítulo
3

Processos para desenvolvimento distribuído de Software

Camila Cunha Borges

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

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.

O processo de desenvolvimento de software, cada vez mais global e distribuído, aumenta a complexidade na construção de sistemas de informação. 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 *software*.

Formatted: Highlight

3.2 Desenvolvimento Distribuído de Software

Grandes investimentos têm permitido uma movimentação do mercado local para o

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 Desenvolvimento Distribuído de *Software* (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 da dos desenvolvedores, clientes e clientes finais. Diferenças culturais incluindo o idioma, tradições, costumes, comportamentos e normas locais.

Formatted: Highlight

Formatted: Highlight

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. [PRIKLADINIKI 2003].

Comment [m31]: Recorrente?

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., Aalgumas empresas buscam soluções em outros países (soluções globais), assim obtendo maiores vantagens competitivas.

Segundo a IDC – *International Data Group* [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, observamos a disponibilidade de profissionais habilitados para trabalhar em outro idioma e incentivos

Comment [m32]: Padronizar siglas? Depois do significado entre parênteses?

Formatted: Highlight

Existem diversas razões para a aplicação do DDS, a seguir apresentamos algumas razões que levam ao desenvolvimento distribuído:

Comment [m33]: Repetido na frase

- **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.

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 freqüência 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:

- **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.

Comment [m34]: Ressaltar que isso não é DDS

- **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

difíceis de acontecerem face a face. Além disso, o fuso-horário dificulta as interações entre os membros das equipes.

- **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 inicio do projeto.

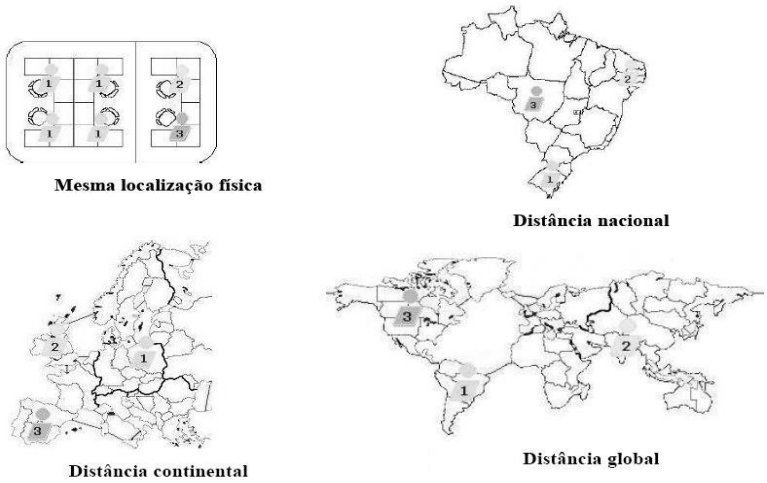


Figura 3.1 – Tipos de Distância Física [PRIKLADINIKI 2007]

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 (*Outsourcing*) para o desenvolvimento de determinados serviços ou produtos de *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, assim podem surgir diversos modelos de negócio. Neste livro abordaremos apenas os modelos de negócio citados nesta seção.

Formatted: Highlight

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 entregue 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 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.

- **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.

3.3 Processos para Desenvolvimento Distribuído de Software

Segundo PRESSMAN [2001], processo é a camada mais importante da Engenharia de Software, fazendo uma ligação entre ferramentas e métodos. Um processo de software é um conjunto de atividades que definem a seqüência em que os métodos serão aplicados e como o produto será entregue, além disso, define os controles que asseguram a qualidade do produto e coordenação das mudanças.

Comment [m35]: Não precisa. O leitor já vai ter lido isso em outros capítulos

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 praticas do desenvolvimento tradicional que podem ser implantadas em um ambiente distribuído e as possíveis adaptações.

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 detalhes antes não percebidos.

A dispersão geográfica agrava os problemas já inerentes à gerência do processo de desenvolvimento. Diferenças culturais, de linguagem, de fuso horário, entre outros aspectos, aumentam a complexidade na comunicação, coordenação e controle durante o desenvolvimento de *software*. Embora já tenha sido formado um corpo de conhecimento sobre desenvolvimento distribuído, ainda existem métodos e técnicas a serem desenvolvidas e amadurecidas.

Comment [m36]: Vem se repetindo isso no texto.

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 [1998]

Comment [m37]: Usar no texto, e não nos títulos

Comment [m38]: Que autor?

Neste trabalho o autor 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):

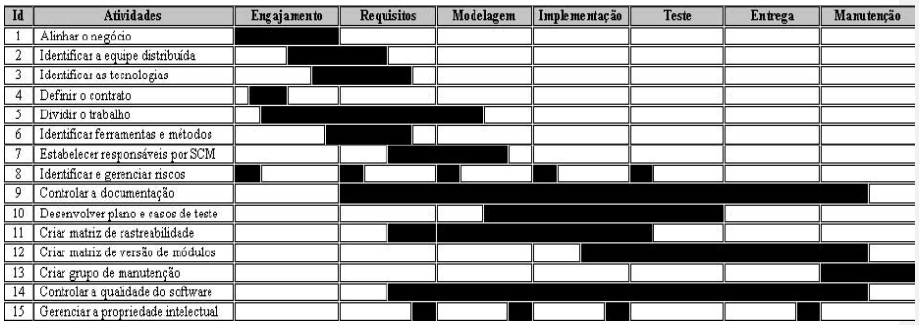


Figura 3.2 – Modelo para projetos DDS [KAROLAK 1998]

A seguir apresentam-se as atividades 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 serem 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 àa 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.
- **Definir o contrato:** Um contrato é um documento que define o escopo do que deve ser feito. Quando o projeto é distribuído este atividade se torna mais

- **Dividir o trabalho:** Após a identificação da equipe, tecnologia e definição do contrato, o autor propõe 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 freqüência 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 identificadas durante a modelagem e implementação.
- **Criar matriz de rastreabilidade:** uma matriz de rastreabilidade é um artefato

Formatted: Highlight

Comment [m39]: Padronização das siglas

Formatted: Highlight

- **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 houverem 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 grande desafio em ambientes DDS. Apesar disso, o uso de metodologias ágeis de desenvolvimento de *software* tem se tornado uma demanda em equipes distribuídas de 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?
Planning Game **Pair Programming** **Continous Integration** **On-Site Custumer**	**Sim. Estes fatores dependem de uma aproximação entre o negócio, cliente e pessoal técnico.**
Small Releases **Metaphor** **Simple Design** **Testing** **Refactoring** **Collective Ownership** **40-Hour Week** **Coding Standards**	**Não. Independem se a equipe é Co-localizada ou não.**

Comment [m41]: Usar software em italic. Definir padrão do livro.

Formatted: Highlight

Formatted: Highlight

Comment [m42]: Discutir como referenciar figuras

Comment [m43]: Traduzir práticas

Conforme apresentado na figura acima, podemos observar que para a utilização do DXP de forma eficaz é necessário que o *Planning Game, Pair Programming, Continous Integration e On-site Custumer* 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.

Formatted: Highlight

Comment [m44]: Como usar separados?

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 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

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.

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, em umno processo de desenvolvimento de uma fábrica de *software* em um ambiente de desenvolvimento distribuído.

Comment [m45]: Seria bom fazer essas referências?

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 ágeis para realizar projetos reais. Os alunos tiveram o período da disciplina (primeiro semestre de 2009) para desenvolver o produto conforme definido no inicio 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 *Bbugtracking,* 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

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 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 auxilio 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 (geralmente duplas) para discutir a sobre o desenvolvimento, em seguida as dúvidas e conclusões eram postadas no grupo de *email*.

Comment [m46]: Pode falar que acontecia pair programming

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 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).

Comment [m47]: Cores não estão visíveis. Traduzir legenda.

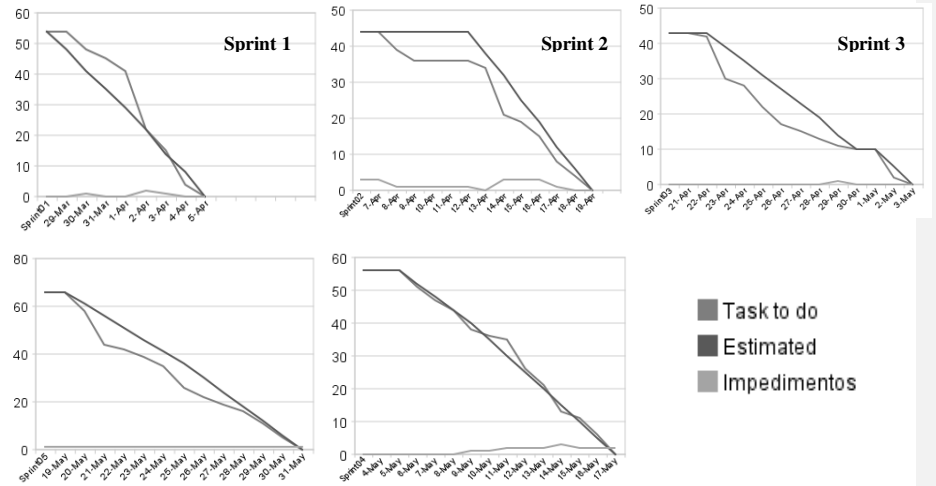


Figura 3.4 – Gráfico *Burndown*

Google docs beta

FireScrum-BugTracking-Projeto

File Edit View Format Insert Tools Form Help

	A	B	C	D	E	F	G
8	BLI2 - Associar um bug a um item de Backlog						
9	Estudo						
10	Projeto	Verificar existencia de tabela com os itens de Backlog		Márcio	CONCLUÍDO	7-Apr	8-Apr
11		Atualizar modelo de dados		Adelnei	CONCLUÍDO	8-Apr	13-Apr
12		Atualizar documento de requisitos		Adelnei	CONCLUÍDO	8-Apr	15-Apr
13							
14	Implementac	Atualizar tela do flex com novos campos e solicitar aprovação do PO	3	Camila / Milton	CONCLUÍDO	13-Apr	14-Apr
15		Atualizar entidade Bug: associar o BacklogItem ao Bug (Java)	1	Ana	CONCLUÍDO	14-Apr	14-Apr

Comment [m48]: Seria bom não ter atividade dividido pra 2 pessoas (regra do scrum).

Figura 3.5 – Planilha de Gerenciamento

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 *Producto 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 *msnMSN*.
- 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 Oportunidades de Pesquisa

Comment [m49]: Seria nesse estilo mesmo?

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 freqüência 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

operação, a relação com outras empresas ou unidades da própria empresa, projetos para equipes distribuídas e outros. Como 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 *Insourcing.*
- **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).

3.6 Considerações finais

3.7 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?

3.8 Recomendações de Leitura

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 boundariesand cultures with high-speed teamwork”. Davies-Black. Palo Alto, EUA, 2001.

Yin, Robert. “Estudo de Caso: planejamento e métodos”. SP: Bookman, 2001, 205p.

Young, C., Terashima, H. “How Did Wr 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.

Sommerville, Ian. Software Enginnering. 8.ed. [S.l] ADDISON WESLEY, 2007.

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/2

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

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, consequentemente, 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

Lista de Figuras

Lista de Tabelas

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

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 a 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.

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) funcional, estrutural e comportamental. Uma especificação é dita como formal quando é baseada em uma linguagem que tem uma sintaxe e semântica bem definidas 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 négocio 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 (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

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)

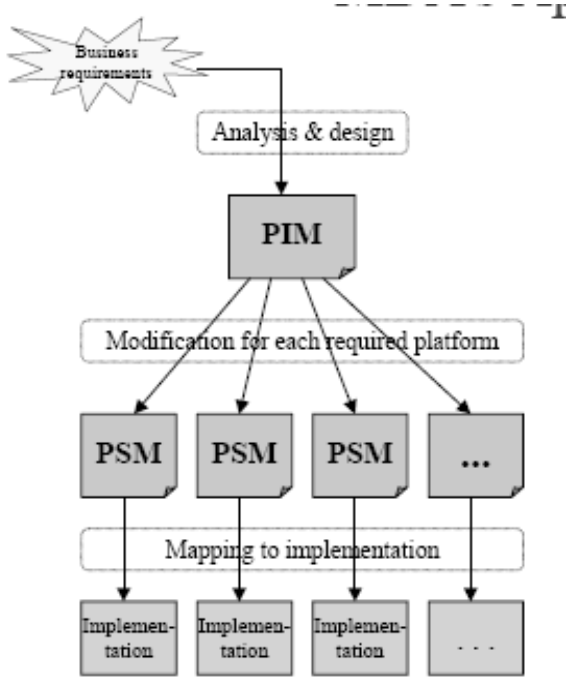


Figura 10.1. Transformações em MDA

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

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

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.

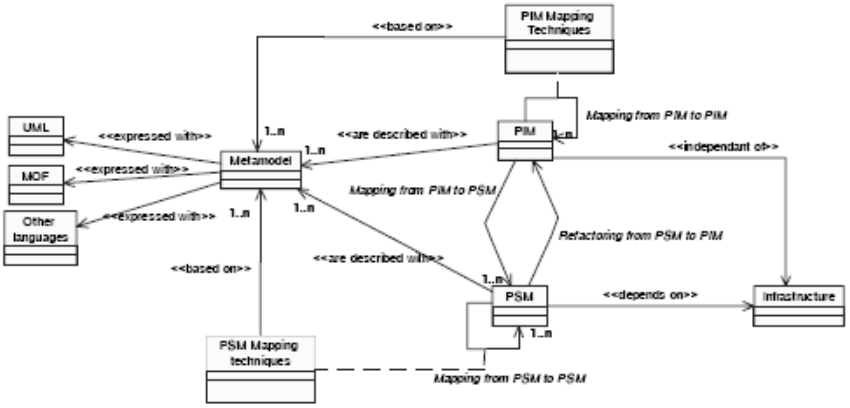


Figura 20.2 Metamodelo MDA

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.

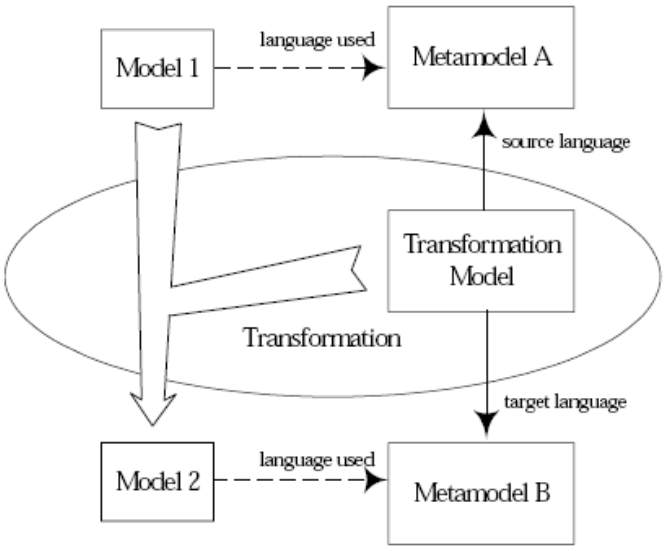


Figura 10.3. Transformações de mapeamentos por metamodelos

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.

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

- 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 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. 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:

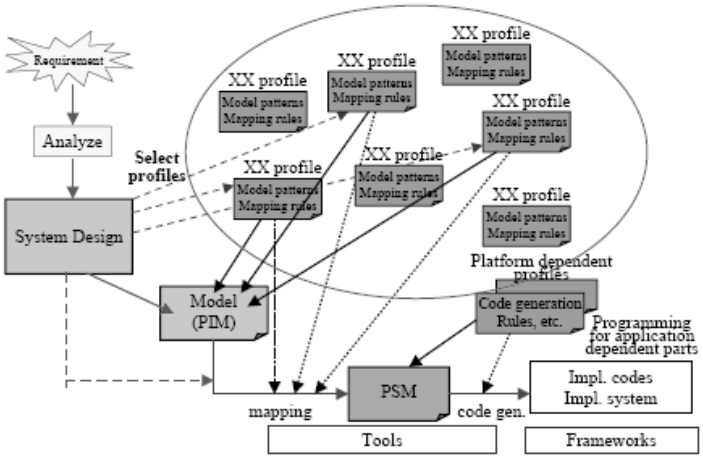


Figura 10.4. Transformações com UML Profile

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.

OMG(standardized)
- UML Profile for EAI (Enterprise Application Integration)
- UML Profile for EDOC (Enterprise Distributed Object Computing)
  - CCA (Component Collaboration Architecture)
  - Entities Profile
  - Events Profile
  - Business Process Profile
  - Relationship Profile
- 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
- 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

Enfim, devido sua imprecisão semântica e complexidade, UML 2.0

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

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.

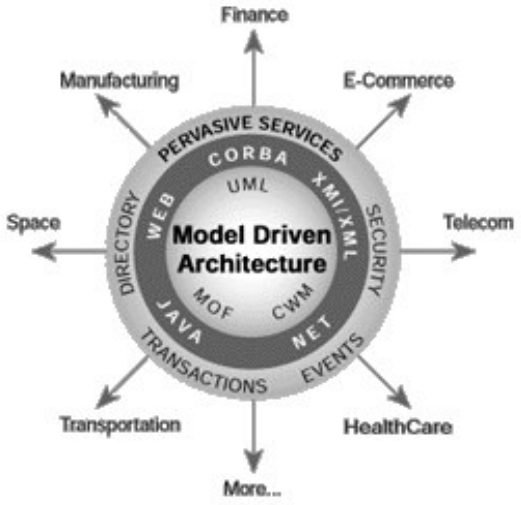


Figura 10.6. Padrões MDA

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

- 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 .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.

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,

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"

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, 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

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.

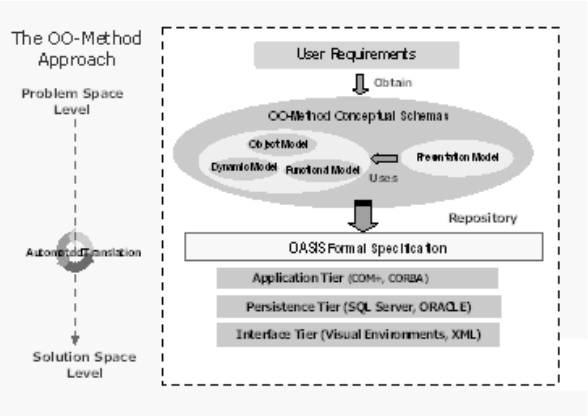


Figura 10.7. Abordagem OO-Method

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 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

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

- 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 e, como base, a linguagem formal OASIS. Isso propicia a validação automática do modelo conceitual a fim de não deixar passar 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
    - 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

- 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 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

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 é 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.

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 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 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 Structs 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

(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 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 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. 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);

- 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 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)s transformação(coes)
- Esboçar documentos de uso, empacotar e 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

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 desses modelos que, segundo o padrão MDA, deve ser alterado?
9. Quais os benefícios em se adotar uma 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).

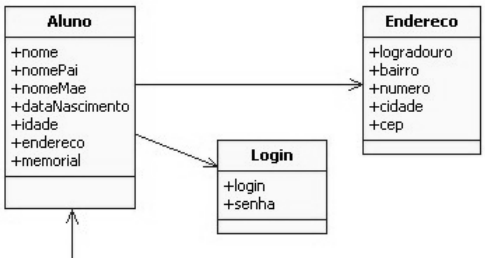


Figura 10.8 Diagrama de Classes

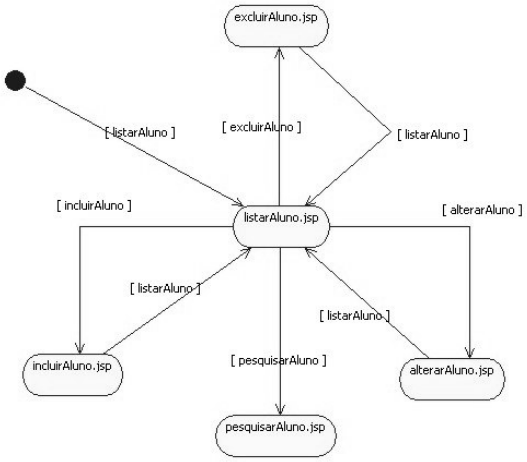


Figura 10.9 Diagrama de Atividades

Referências

MDA productivivity 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 Aproach. 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.

Capitulo

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.

Comment [A50]: Formatar Tabulação para 1,27cm (ver os demais parágrafos)
- Também na formatação de todos os parágrafos, o **espaçamento antes** é de 6 pontos

→RETIRAR ESTE ESPAÇO EM BRANÇO

1.1. O que é Modelagem de Processos

→RETIRAR ESTE ESPAÇO EM BRANÇO

→RETIRAR ESTE ESPAÇO EM BRANÇO

Comment [A51]: Não devem existir espaços (linhas em branço) entre as seções e/ou parágrafos, basta ajustar a formatação do parágrafo para ter **espaçamento antes** de 6 pontos

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 elementos destas áreas funcionais. Durante este trabalho, os participantes apresentam um aumento do entendimento do negócio.

Comment [A52]: Não existe esta vírgula

Comment [A53]: Não seria melhor substituir elementos por “pessoas” ou “ indivíduos” ?

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 / eficaz, ou mesmo antecipar sua complexidade, redundâncias e não conformidades (problemas). Se o processo é alguma coisa nova que a empresa está planejando executar, o modelo pode ajudar a assegurar sua eficiência desde o início.

Comment [A54]: 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 (claros), são a chave para a comunicação.

Comment [A55]: Não entendi ! Os processos são claros ou claros está significando a oração intercalada " , é claro ,"

→RETIRAR ESTE ESPAÇO EM BRANÇO

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.

→RETIRAR ESTE ESPAÇO EM BRANÇO, IDEM TODO RESTANTE DO TEXTO.

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. Há dois tipos principais de modelos de processo de negócio:

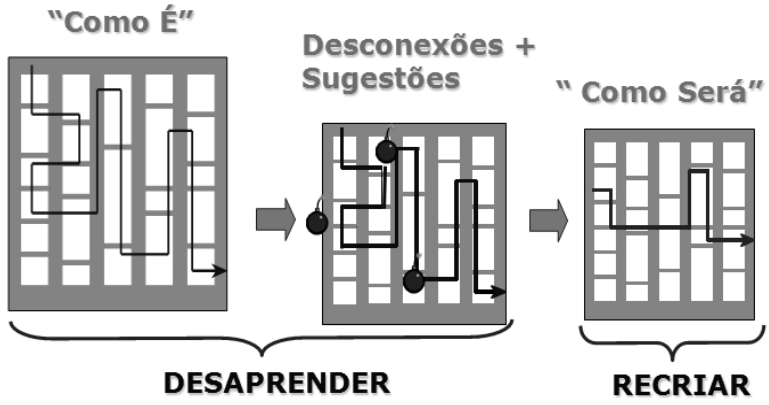
- Modelo as is or baseline (a situação atual);
- Modelo to be (a nova situação pretendida).

Comment [A56]: Sugestão entre aspas e negrito :
"**as is** ". Idem para "**to be**"

*falar um pouco mais sobre os modelos acima.

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 [A57]: Não tenho certeza, mas acho

2.1. Objetivo da Modelagem de Processos Software

Comment [A58]: Suponho que esta deva ser a segunda seção do capítulo, ou seja, 16.2 . Portanto, deveria ser com fonte tamanho 13

3.1 Vantagens e Desvantagens da Utilização de Modelagem de Processos

Comment [A59]: Fonte 13, seção 16.3

3.1.1 Vantagens

Comment [A60]: 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 ..

4.1. Linguagem de Modelos de Processos

Comment [A61]: Fonte 13, seção 16.4 . Fazer um parágrafo simples de apresentação da seção, antes de falar da linguagem.

4.1.1. BPM

Comment [a62]: Esta seria a seção 16.4.1 do capítulo.

O que é **BPM** segundo esses autores**:**

Comment [A63]: Eu reescreveria da seguinte forma:
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 [A64]: Referência incorreta, falta o **ANO** de publicação . Idem para as demais do Capítulo

Business Process Modeling, ou Modelagem de processos de negócio, 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 [A65]: A Sigla BPM é mais conhecida no mundo commercial/acadêmico como acrônimo para Business Process Management. Assim , seria melhor dizeres que BPM , envolve modelagem, execução, monitoramento e análise de processos de negócios e, nesse contexto , "Modelagem de processos de negócios, é o conjunto ....."
Enfim, eu retiraria essa expressão em inglês.

Comment [A66]: Referência não está conforme padrão SBC. Hà outras referências erradas como essa ao longo do capítulo, corrigir !

Figura 2 (BPM)

Comment [A67]: - 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

enquanto que as funções, que em conjunto descrevem a missão da empresa, são verticais (ex.: contabilidade, vendas, logística).[1]

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

Comment [A68]: - Fonte de figuras deve ser Helvética 10

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

Comment [A69]: Falar que em 2005 BPMI se funde com o grupo OMG, e BPMN agora está na

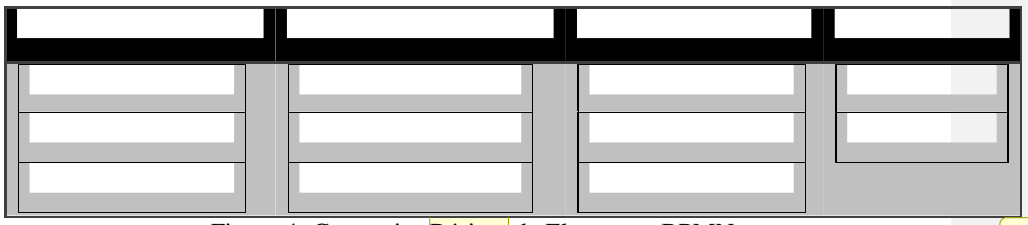


Figura 4- Categorias Básicas de Elementos BPMN

Comment [A70]: - Fonte de figuras deve ser Helvética 10
** Idem para todas Figuras do Texto.

Comment [A71]: Dica:
Colocar Evento, Atividade e Gateway com um texto da figura , suprimindo o tópico:
•Evento, Atividade e Gateway

IDEM RESPECTIVAMENTE PARA OS PRÓXIMOS ITENS 2 e 3.

Comment [A72]: Colocar um texto para esta figura.

Comment [a73]: Colocar um exemplo de processo modelado com BPMN

Tabela 1. Ícones e estereótipos do SPEM.

AQUI

Comment [A74]: O texto para tabelas fica no topo da mesma(ver template SBC) e a fonte é helvética 10.

Comment [a75]: COLOCAR UM EXEMPLO SIMPLES DE PROCESSO MODELADO COM SPEM, pode ser o RUP, o mesmo de um material enviado pelo Prof.

- **Category:** serve para associar um pacote a um elemento de outro pacote;

Comment [a76]:
Retirar esse quadrado após o ponto do tópico.
IDEM para os próximos tópicos.

4.2.1 OMG(Object Management Group)

Comment [a77]: Ver numeração da seção que seria 16.4.2

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.

Comment [a78]: Incluir também neste parágrafo SPEM e BPMN , vistos nas seções anteriores, como padrões da OMG.

6.1. Ferramentas de Modelagem *(colocar exemplo didatico)

Comment [a79]: Numeração da seção incorreta.. A correta seria 16.5 .
Renumerar todas seções posteriores do capítulo: 16.6, 16.7 , 16.8 ,ETC.

Business Process Management “Concepts, languages, Architectures”,

Comment [a80]: Colocar texto descritivo , caso o leitor queira se aprofundar no assunto , para cada r sugestão de leitura

Referencias

[2] http://www.bpmn.org

Comment [a81]: Retirar os colchetes das referencias e para cada referencia web, complementar com "acessado em ..... "

- Também colocar as referencias em ordem alfabética.

Comment [a82]: Referencias [1] e [2], [15] colocar no padrão, p exemplo: BPMN , HTTP://www.bpmn.org, Acessado em outubro 2009.

Flávia Leite Soares, Willame Pereira

<Incluir aqui, antes da primeira seção do capítulo, uma visão geral do capítulo.>

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 está inserida, os fornecedores, os acionistas e, principalmente, seu corpo funcional.

Comment [rbsa83]: Padronizar os modos de referenciar…

Comment [rbsa84]: Tá meio estranho… talvez usar "em" em vez de "os"

Comment [rbsa85]: crase

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”.

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.

Comment [rbsa86]: usar o padrão de referencia

Comment [rbsa87]: ponto depois das aspas

Comment [rbsa88]: usar o padrão de referencia

Comment [rbsa89]: alguns

Comment [rbsa90]: “por exemplo”

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”.

Comment [rbsa91]: usar o padrão de referencia

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):

Comment [rbsa92]: usar o padrão de referencia

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;

Comment [rbsa93]: padronizar referencia

Comment [abv94]: Seria interessante colocar uma "nota de rodapé" informando o que quer dizer esse "aporte de conhecimento"

Figura 6.1. blabla

Comment [abv95]: Faltou o nome da figura. E, conforme informado pelo template, as figuras devem ser em escala cinza.

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 por Deming 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 estas estão aptas a sobreviver nesse novo cenário mundial de concorrência.

Comment [rbsa96]: Padronizar referencia.

Comment [abv97]: "estas" está se referindo a que? Seria à "qualidade ideal"
Acho que o texto pode ser escrito de forma mais clara.

Figura 6.2. Bla bla bla

Comment [abv98]: Nome da figura? Colocar em escala cinza.

O componente informação é de suma imnportância nesse cenário de competitividade, pois ela é a matéria-prima necessária à criação de conhecimento, um instrumento essencial para:

Comment [rbsa99]: Talvez poderia falar isso de outra forma, ou destacar a palavra "informacao"… algo do tipo.

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).

Comment [V100]: Acho que deveria traduzir nesse momento

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. Existem algumas abordagens para o Controle da Qualidade Total:

Comment [rbsa101]: Como as abordagens estão apresentadas em topicos, melhor tirar o ":" e dizer que elas serão apresentados nos sub-topicos abaixo.

Reconhecido mundialmente como o grande promotor do Controle da Qualidade no Japão, deixou grandes contribuições para o desenvolvimento da qualidade. 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:

Comment [rbsa102]: Ficou meio estranha essa referencia da forma que ta (ta sem ligação)... melhor ver um outro modo de colocar.

1) Criar uma constância de propósitos de melhorar produtos e serviços.

Comment [rbsa103]: Padrozinar pontuacoes dos marcadares… geralmente ta utilizando ";" e aqui ta usando "." Nos itens de 1 a 13.

14) Criar um estrutura na alta administração que tenha como função implantar os 13 pontos anteriores. A transformação é tarefa de todos.

Comment [rbsa104]: uma

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."

Comment [rbsa105]: ajustar espaçamento conforme padrao do SBC.

6.4.3. Crosby

Comment [rbsa106]: Daqui pra baixo precisa dar uma revisada no texto pra ajustar o espaçamento conforme o padrao do SBC.

Para sedimentar sua filosofia Crosby (1985) instituiu seus 14 pontos, que constituem as etapas de implementação de sua abordagem, são eles:

Comment [rbsa107]: colocar uma virgula

Comment [rbsa108]: Acho que aqui necessita de uma pausa mais longa, ou “;” ou “.” talvez.

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;

Comment [rbsa109]: Colocar espaço para padrozinhar com as numerações acima... ajeitar também nos itens 11 a 14.

A abordagem de Ishikawa nasceu a partir da compilação de diversos aspectos do trabalho de vários especialistas como Deming, Juran e Shewart, acrescendo a eles uma grande preocupação com a participação do elemento humano e trazendo para o controle da qualidade uma visão humanística sob a influência dos trabalhos de Maslow, Herzberg e McGregor.

Comment [rbsa110]: É bom referenciar nomes.

Sua filosofia é voltada para a obtenção da qualidade total (qualidade, custo, entrega, moral e segurança) 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.

Comment [rbsa111]: Acho que um "desde" aqui fica legal.

- Voluntarismo: os círculos devem ser criados em bases voluntárias e não por ordens superiores.

Comment [rbsa112]: Padronizar com ";"

- Eventual participação total: os círculos precisam estabelecer como seu objetivo final a participação total de todos os trabalhadores do mesmo local de trabalho.

Comment [rbsa113]: Não entendi essa frase. Verificar se está OK.

Comment [abv114]: Referenciar!

- 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;

Comment [abv115]: Seria interessante indicar a seção

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.

Comment [abv116]: Referenciar (http://www.eps.ufsc.br/disserta/fiates/cap3/cap3.htm)

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 por último com a comunidade na qual está situada. O quadro abaixo mostra como esta interação pode ocorrer:

Comment [rbsa117]: Melhor colocar: " A Tabela 6.1 mostra…"

Tabela 6.1. Satisfação das pessoas da empresa.

Comment [rbsa118]: Seguir padrão de letra do SBC

- Parte do reconhecimento das necessidades das pessoas e estabelece padrões para o atendimento destas necessidades.

Comment [rbsa119]: Padronizar com ";"

Para lshikawa (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.

Comment [rbsa120]: "." Depois das aspas.

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 lshikawa.

Comment [rbsa121]: Vai seguir a nova regra gramatica???

Comment [rbsa122]: É bom sempre referenciar quando citar nomes.

O que fazem é analisar a demanda dos consumidores, avaliar a qualidade do serviço e, com estas informações, definir o que deve ser medido, melhorado e garantido.

pode ser utilizada para a indústria de serviços, tomando-se o cuidado de fazer algumas adaptações quando necessárias.

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 (Figura 2.1) e tem o seguinte significado:

- 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 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.

- 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.

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).

Comment [rbsa123]: Colocar uma virgula

Comment [rbsa124]: Que figura é essa? Ajeitar também a numeração do capitulo... deve ser Figura 6.alguma coisa.

Comment [rbsa125]: Parte de vermelho tá repetida.

Comment [rbsa126]: Colocar em italico.

Comment [rbsa127]: Seria bom alguma frase aqui introduzindo a equeçao de baixo, porque do nada aparece a equeçao.

3.1. **Orientação pelo cliente**: Produzir e fornecer serviços e produtos que sejam definitivamente requisitados pelo consumidor.

Comment [rbsa128]: Tá estranho ":" e depois ":" de novo.

Deming mostra como as coisas acontecem em uma reação em cadeia quando o foco da empresa está na qualidade.

Comment [rbsa129]: É bom dizer aqui que isso é mostrado na figura 6.3.

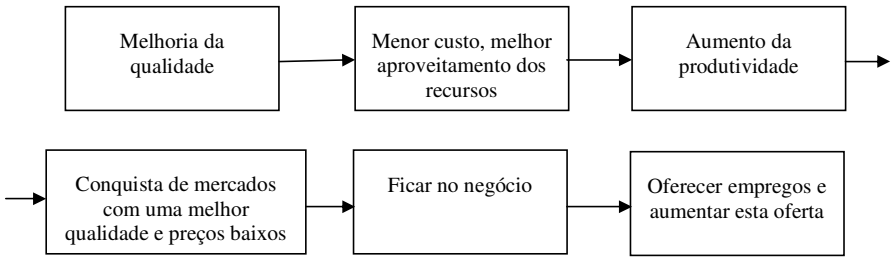


Figura 6.3. Cadeia Competitiva da Empresa. Walton (1989).

Comment [rbsa130]: Seguir padrão de letra do SBC

Formatted: Font: (Default) Helvetica, 10 pt, Bold

Ainda hoje existem administradores que têm um conhecimento ilimitado sobre quase todas as coisas. Estes gênios 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, supervisares, 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 freqüência adequada de coleta. A partir destes dados, uma análise com base em técnicas estatísticas é que levará a resultados satisfatórios.

Comment [rbsa131]: crase

Comment [rbsa132]: supervisores

Comment [rbsa133]: virgula

7.5. **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).

Comment [rbsa134]: maiusculo para padronizar.

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.

Comment [rbsa135]: Hifen.

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

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, deve-se tomar as providências necessárias para manter os processos dentro de níveis aceitáveis de variabilidade.

Comment [rbsa136]: plural

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).

Comment [rbsa137]: Pausa mais longa, melhor usar "." ou ";".

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.

Comment [rbsa138]: Novamente, pausa mais longa.

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.

Comment [rbsa139]: Esse trecho em vermelho fica melhor entre virgulas

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.

Comment [rbsa140]: plural

Os erros mais freqüentes na elaboração do formulário são:

Comment [rbsa141]: vai seguir a nova regra gramatical e tirar o trema???

Os dados podem ser coletados através questionários, folhas de verificação,

Comment [rbsa142]: de

Figura 6. 4. Folhas de coleta de dados

Comment [rbsa143]: Tem que introduzir a figura e colocar alguma explicação sobre ela!!!

Comment [rbsa144]: Seguir padrão de letra do SBC

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).

Comment [rbsa145]: virgula

Comment [rbsa146]: virgula

Figura 6. 5. Problemas x Impactos demonstrado pelo diagrama de Pareto.

Comment [rbsa147]: Tem que introduzir a figura e colocar alguma explicação sobre ela!!!

Comment [rbsa148]: Seguir padrão de letra do SBC

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.

Comment [rbsa149]: Acho que ficaria melhor assim: ... dessas causas, eliminando, dessa forma, uma parcela...

Este diagrama, também chamado de diagrama de lshikawa 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.

Comment [rbsa150]: O verbo de vermelhor deve estar no singular: ser

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.

Comment [rbsa151]: vai seguir a nova regra gramatical?

Comment [rbsa152]: vai seguir a nova regra gramatical?

Exemplo do digrama Ishikawa – possíveis causas do atraso entre uma cirurgia e outra.

Comment [F153]: Será usado outro exemplo de software

Comment [rbsa154]: Colocar a legenda no padrão e mais uma vez introduzir a figura e explica-la.

Esta técnica é utilizada para representar seqüencialmente 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.

Comment [rbsa155]: Regra gramatical??

Figura 6. 5. Número de correções feitas no software por semana.

Comment [rbsa156]: Colocar na formataçao padrão e novamente introduzir e explicar.

1.7.1. Gerenciamento por Processos

Comment [V157]: Aqui não seria 6.7.1? Vocês ainda vão escrever?

1.8. Garantia da Qualidade

Comment [rbsa158]: 6.8

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 (1992), como a "embaixatriz" do cliente na empresa.

Comment [V159]: Da empresa? Acho Garantia da qualidade é um conceito mais genérico. Não deveria restringir à empresa.

Comment [V160]: Independente da concorrência é preciso ter qualidade. Acho que pode tirar essa frase.

Comment [V161]: Verificar a padronização das referências.

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 (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

A garantia da qualidade busca o “defeito zero”, 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 do elemento humano não se pode atingir a garantia da qualidade.

A qualidade é garantida pela condução 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 conduzida uma auditoria da qualidade para verificar se todas as atividades da qualidade estão sendo conduzidas como planejado.

1. **Planejamento da qualidade**. Por meio de contato direto com o consumidor final, 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.

O ciclo da garantia da qualidade começa no cliente. Por meio de uma pesquisa do 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 verificadas 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, voltando-se 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]. As auditorias praticadas em controle da qualidade devem sempre ter como objetivo ajudar as pessoas, cooperando com elas no sentido de trazer

Comment [rbsa163]: Isso ja foi explicado em outra parte desde capitulo, anteriormente.

Comment [V164]: Acho melhor substiruir por: das pessoas. Elemento humano soa estranho

Comment [V165]: Está repetindo essa palavra.

Comment [V166]: Acho que pode tirar o final, já que consumidor é o final.

Comment [rbsa167]: virgula

Comment [V168]: Não seria pesquisa de Mercado?

Comment [V169]: Substituir por alinhadas

Comment [V170]: Substituir por retornando.

Comment [V171]: Ficou meio vaga essa

(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 a garantia da qualidade é (conforme conceito japonês) garantir a satisfação do cliente por um longo tempo a um preço que este possa comprar (que significa baixo custo) 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, no local 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. 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].

Como foi já foi visto, para que a qualidade seja garantida numa organização, todos os processos devem garantir a qualidade para o processo seguinte. Então, o relacionamento da empresa com seus clientes (vendas) e da empresa com seus fornecedores (compras) também devem ser norteados pelo princípio básico de satisfazer as necessidades do cliente.

O setor de vendas, dentro desse contexto de satisfação das necessidades do cliente, não deve somente "aceitar 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".

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 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

Comment [V173]: Colocar as letras minúsculas

Comment [V174]: É melhor colocar entra vírgulas

Comment [V175]: Não precisa desse texto.

Comment [V176]: Já foi falado anteriormente

Comment [V177]: Local?

Comment [V178]: Acho que poderia explorer mais essa explicação. Ficou um pouco vaga.

Comment [V179]: Consertar

Comment [V180]: Repetindo as palavras

Comment [V181]: Explicação confusa

Comment [V182]: Acho melhor receber pedidos

Comment [V183]: Por não se ter qualidade, não é?

Comment [rbsa184]: Tabela 6.2

Tabela 6.2. Exemplos de Itens de Controle nos Processos de Marketing.

	1. Educação e treinamento do pessoal de marketing em Controle de Qualidade. 2. Plano de vendas (obtenha precisão nas projeções, análise das projeções).
	3. Educação e treinamento do usuário do produto ou serviço.
	4. Taxa de defeituosos no estoque.

Comment [rbsa185]: Padronizar a formatacao da letra de acordo com o SBC.

Comment [V186]: É melhor dos responsáveis

Comment [V187]: 1.É melhor: precisão e análise das projeções.

Comment [V188]: Acho melhor deixar só treinamento

Comment [V189]: Faltou a palavra produtos

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 melhoria da 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 com os fornecedores. Infelizmente, muitas empresas os maltratam, mudando suas programações de compra e prazos de pagamento ao sabor de suas necessidades, jogando muitas pequenas e médias empresas brasileiras ao desespero, à concordata e à falência.

Comment [V190]: Acho que em vez de melhoria, já deve colocar boa

Comment [V191]: Maltratam quem?

Comment [V192]: Jogando ao desespero soa estranho

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. O consumidor, ao comprar um produto de uma empresa, está na verdade comprando de uma "cadeia de empresas". Então é necessário que todas busquem a qualidade de tal forma a tornar a "cadeia competitiva" [Campos 1992].

Comment [V193]: Tirar essa virgule e colocar depois do está

Comment [V194]: Está estranho essa frase final

1.10. Implantação do TQC

Comment [V195]: Atualizar o número

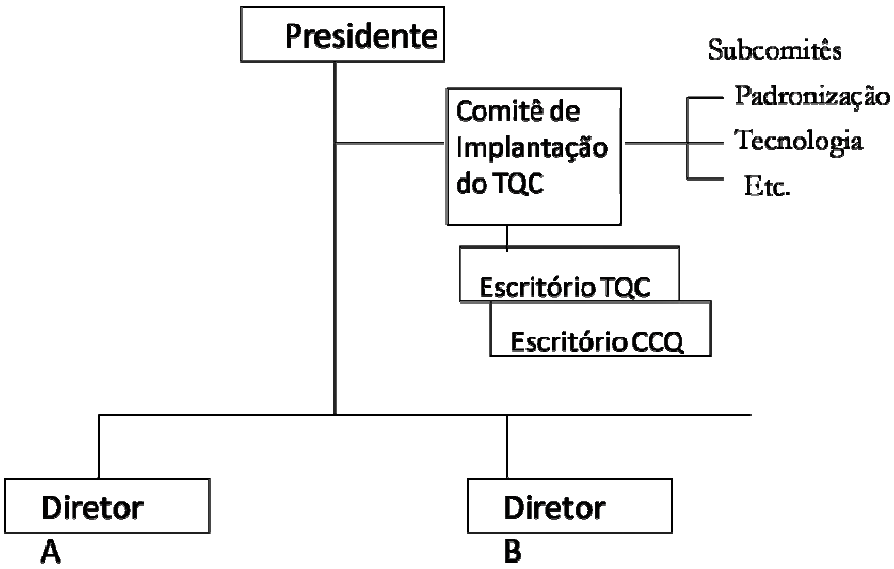


Figura 6.6. Organização para Implantação do TQC. Adaptado de [Campos 1992].

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 a este como ao consultor externo.

Comment [V196]: Onde está o coordenador?

Comment [rbsa197]: Padronizar de acordo com SBC

Comment [V198]: Ajustar a concordância desta frase.

Comment [V199]: Colocar Coordenação

- Monitora todo o processo de implantação do TQC.
- Avalia o estado atual e relata mensalmente ao Comitê de Implantação do TQC.
- Difunde os resultados do TQC por toda a empresa.

Comment [V200]: Monitoramento

Comment [V201]: Avaliação

Comment [V202]: Relato mensal

Comment [V203]: difusão

Figura 6.7 - Gerenciamento da Implantação do TQC. Adaptado de [Campos 1992].

Comment [rbsa204]: padronizar

6.12. Exercícios

Comment [V205]: Eu achava que o exercício estava pronto. Estamos com a versão correta?

Walton, Mary. (1989) “O método Deming de administração”, Marques Saraiva, Rio de Janeiro.

A utilização de normas foi algo que teve constantes evoluções ao longo dos anos. Desde a idade média, os filósofos padronizavam medidas e cálculos nos primeiros documentos relacionados a padrões técnicos. A idéia de um conceito documentando-o e 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 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.

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, SOARES, 2007].

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 à 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,FERREIRA, 1988].

Comment [C206]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

Comment [C207]: Faltou a palavra ser. "Pode ser obtida…".

Comment [C208]: Referência for a da formatação.

Comment [C209]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

Comment [C210]: Referência for a da formatação.

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 14000[1].

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 administrável 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 países[2] 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.

Comment [C211]: Faltou o espaço depois do 50.

Comment [C212]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

Comment [C213]: A palavra "administrável" está redundant, haja vista que já tem a palavra "controle".

No campo da tecnologia, grande parte das normas publicadas está subsidiada a parcerias realizadas com o *International Eletrotechnical Comission* (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:

Comment [C214]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

Comment [C215]: Retirar o sinal de pontuação antes da palavra "Informação"

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 Committe 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 x ilustra a atual hierarquia formada pela ISO, IEC e JTC1.

Comment [C216]: Faltou o número da figura

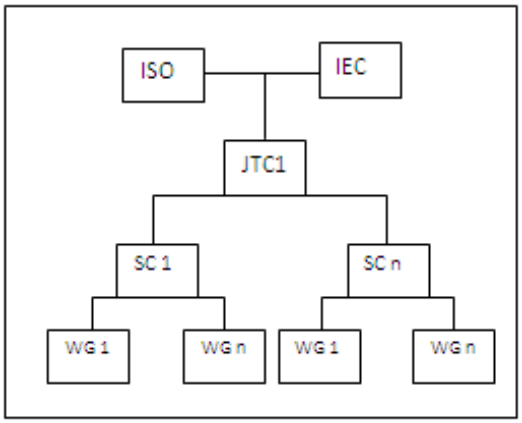


Figura x: Estrutura ISO/IEC/JTC1

Comment [C217]: Faltou o número da figura.

Fonte: Adaptado de (Koscianski e Soares 2007)

Comment [C218]: Referência for a da formatação.

Observando a Figura X 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.

Comment [C219]: Faltou o número da figura.

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.

Comment [C220]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

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 cínico (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 [C221]: Faltou espaço depois da palavra “exterior”.

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:

Comment [C222]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

Camfield e Godoy (2003) afirmam ainda que as normas da série ISO 9000

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 X apresenta a família ISO atual:

Comment [C223]: Faltou o número da tabela

Comment [C224]: Faltou o número da tabela

Tabela X: A família ISO 9000

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].

Comment [C225]: Referência for a da formatação.

Comment [C226]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

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 provessem 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 x).

Comment [C227]: Faltou o número da figura.

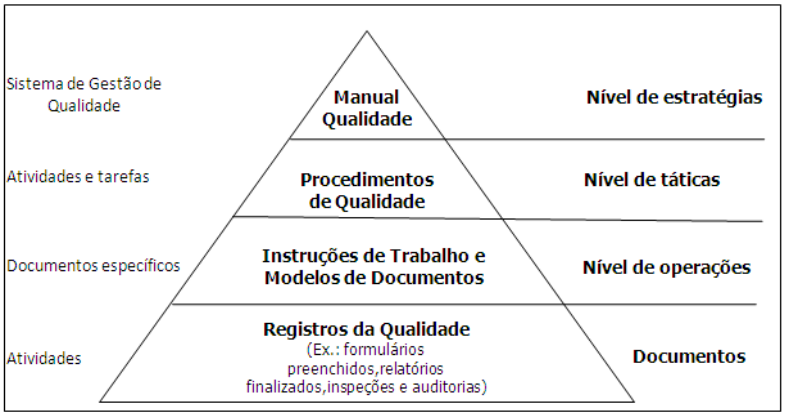


Figura x: Requisitos de documentação para uma organização

Comment [C228]: Faltou o número da figura.

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 as exigências impostas pelos guias de referência da norma desenvolvendo

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.

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.

Comment [C229]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

O termo certificação inflige características bem perplexas em seu significado. Ferreira (2004) adota certificação como o convencimento da verdade ou da certeza de algo, tornando ciente daquilo que está se abordando. Para a ISO, o pensamento não abordou aspectos diferentes considerando os padrões que precisam ser mantidos no desenvolvimento de suas certificações.

Comment [C230]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

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 chaves[3] 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, STROMBERG, 2003]. Melloti, Darlan Jader et. al. (2007) descreve que esta norma possuía uma visão desmembrada de negócios para

Comment [C231]: Referência for a da formatação.

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.

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 x) da ISO 9001:2000 tais como:

Comment [C232]: Faltou o número da tabela

Comment [C233]: Faltou o número da tabela

Tabela X: Elementos-Chave da ISO 9001:2000

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, Carlos Henrique Pereira 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:

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, STROMBERG, 2003].

Comment [C234]: Referência for a da formatação.

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, Carlos Henrique Pereira et. al. 2009].

Comment [C235]: Referência for a da formatação.

Comment [C236]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

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 176[4], 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 (2008) 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 gestão[5]

- 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.

Comment [C237]: Este trecho deve ser entre vírgulas

[4] "Comitê técnico *Quality managements and quality assurance* (ISO/IEC 176), subcomitê *Quality systems* (SC 2),conforme a ISO/IEC Guide 21-1:2005" [ABNT,2008,p. v]

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, Carlos Henrique Pereira et. al. 2009].

Comment [C238]: Referência for a da formatação.

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.

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 aplicação da certificação ISO 9001 com o propósito de normatizar e qualificar a gestão da qualidade baseado nas chamadas *Fábricas de Software*[7].

Comment [C239]: Como este é o trecho inicial deste tópico não há espaçamento de parágrafo, conforme o o próprio template sugere.

- 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

impostas através da elaboração de direitos e deveres utilizados pelos compradores e desenvolvedores relativos à instalação dos sistemas.

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

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 aplicados durante o desenvolvimento do software. Os mais conhecidos são: ISO/IEC 15504, CMMI e MPS.BR.

Com essa evolução, conceitos importantes surgiram gradativamente, tais como divisão de software, arquitetura *top-down* e *botton-up*, diagramas e modelagens, nos conduzindo 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.

Comment [W240]: Primeira pessoa do plural? Substituir por 3º. "Conduzindo a engenharia de software ao estado atual"

Os modelos de qualidade foram criados para ser um guia destinado a melhorar os processos organizacionais e a habilidade deste 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 Engeneering 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, tornando-se um domínio específico da computação para a avaliação de uma empresa de desenvolvimento de sistemas.

Comment [W241]: Deste quem? Processos organizacionais? Então deveria ser "Destes".

Comment [F242]: Frase muito extensa, deveria ser dividida. Uma frase a té RUP e em seguida: Esse modelo de maturidade tornou-se um domínio específico...

O CMMI procura nortear a organização no sentido de implementar a melhoria contínua do processo de software, e o faz através de um modelo que contempla duas representações, divididas em níveis, priorizando de forma lógica as ações a serem realizadas. Quanto maior o nível, maior a maturidade da organização, o que se traduz em maior qualidade do produto final, com maior previsibilidade em cronogramas e orçamentos.

Comment [W243]: Melhor dizer que "PODE se traduzir em maior qualidade…" ou "Geralmente se traduz...". Até pq sabemos q qualidade do processo não garante 100% qualidade do produto.

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 ou de maneira contínua, onde é mensurada a capacidade em práticas individuais, conforme ilustrado na Figura 3.1.

Comment [F244]: Poderia ficar: maturidade (avaliada por estágios da organização)

Comment [W245]: Não entendí. "idéia de maturidade ou de maneira contínua?" Ah, sim, faltou a vírgula depois de maturidade. Na verdade, é "em etapas consecutivas ou de maneira contínua".

Comment [W246]: Acho melhor fazer a associação "nível de maturidade e estagiada" e "nível de capacidade e contínua" aqui. Lá na frente só aprofunda. Acho melhor pro entendimento.

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 maturidades superiores deverá cumprir metas, chamadas áreas de processo (*Process Area – PA*). As *PAs* são estáticas e funcionam como uma coleção de práticas que representam o nível de maturidade.

Comment [W247]: Meta é sinônimo de area de processo? Uma área de processo possui diversas metas. Não?

Comment [F248]: Compreendidas em cada área de processo. (Sugestão)

d) **Fontes de Aquisição.** Atua na aquisição de produtos sempre que o projeto

Comment [F249]: Essa tradução é official? Conheço o termo “Gestão de Fornecedores” também.

Poderia-se mostrar uma visão geral dos componentes do modelo:

Comment [F250]: Sugestão

O CMMI pode ser representado de forma "contínua" ou "por estágios", permitindo que a organização disponha de diferentes caminhos para a melhoria dos seus processos de acordo com seu interesse.

Comment [W251]: Com a sugestão que eu dei acima, o leitor já leria isso e teria uma noção do que seja "contínua" e "estágio". Assim, não precisaria apresentar isso como se tivesse aparecendo pela 1º vez no texto.

Um caminho permite que as organizações melhorem de forma incremental os processos correspondentes a uma ou mais *PAs*, que são selecionadas pela empresa de forma individual. O outro caminho permite que as organizações melhorem um conjunto de processos interrelacionados e, de forma incremental, tratem sucessivos conjuntos de *PAs*. Esses dois caminhos de melhoria associam-se aos dois tipos de representações: o primeiro refere-se à representação contínua, na qual é associada a expressão "nível de

Comment [W252]: Melhor dizer, "A representação continua permite que…".

Comment [W253]: Melhor colocar ponto e começar em novo período. Lí várias vezes pra entender. Dica: "A própria empresa seleciona em que áreas de processo ela será avaliada."

Comment [W254]: Melhor dizer: "A representação estagiada, por sua vez, permite que...".

Representação por Estágios

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.

Comment [W256]: É bom dizer que o foco aqui é a organização como um todo, assim como será dito na seção "Representação contínua" que o foco dela é numa área de processo.

Comment [F257]: Concordo com W16, deveria ter uma frase inicial deixando isso claro.

Comment [W258]: No CMMI-Dev tem uma figura mostrando isso. Seria bom colocar ela.

b) **Nível 2 – Gerenciado.** Neste nível, os projetos da organização têm a garantia de que os requisitos são gerenciados, planejados, executados, medidos e

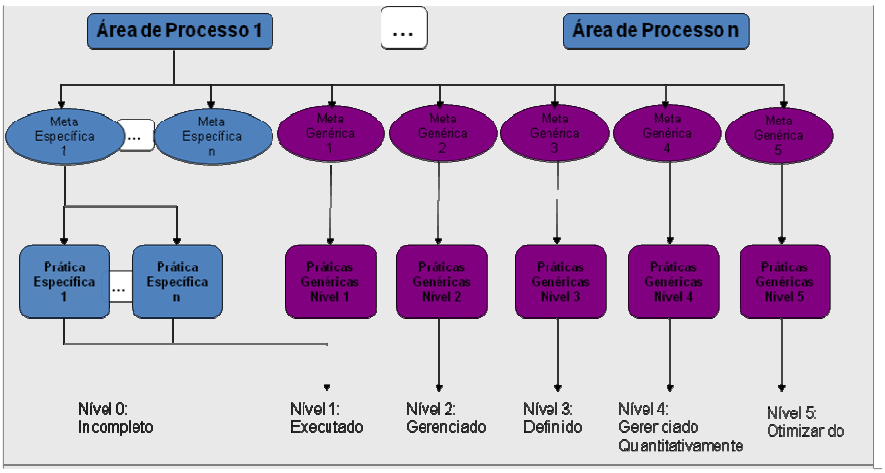
controlados de acordo com o planejado. O gerenciamento de projetos é o foco principal deste nível.

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, às 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.

Comment [W259]: Virgula importante faltando.

Comment [W260]: Crase?

Comment [F261]: Ta errado mesmo…



Comment [F262]: Sugestão de figura

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 3.3.

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.

Comment [W263]: Melhor dizer, “nível 2”. Melhor entendimento nesse caso.

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.

Método de Avaliação do CMMI (SCAMPI)

Comment [W265]: Tem muita coisa sobre Scampi!!! Acharia melhor terem falado sobre a arquitetura do CMMI e das constelações lá na introdução, do que falar tanto sobre Scampi aqui.

O SCAMPI é um método onde as validações são aceitas se as mesmas forem baseadas em critérios conhecidos e consequentemente haja uma contextualização da informação organizacional com o CMMI.

Comment [W266]: Não entendí? Tá meio sem sentido esse "e consequentemente haja..."

O SCAMPI procura ajustar seu método a finalidade e necessidades da organização, ou

parâmetros que devem ser documentados no plano de avaliação com o objetivo de respaldar sua execução.

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

motivações tipicamente externas para as organizações cristalizarem o processo de melhoria [Almeida 2007].

Tabela 3.1. Fase 1 - Plano e Preparação para a Avaliação [Almeida 2007]

Tabela 3.2. Fase 2 - Administrando a Avaliação [Almeida 2007]

Tabela 3.3. Fase 3 - Relatório do Resultado [Almeida 2007]

MPS.BR

O Modelo MPS.BR estabelece não somente um modelo de processos de software, mas também um processo e método de avaliação de processos, além de um modelo de negócio para subsidiar as empresas brasileiras que desenvolvem software. O MPS 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 3.3) [SOFTEX 2009].

Comment [W267]: "um método…"

A Figura 3.4 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).

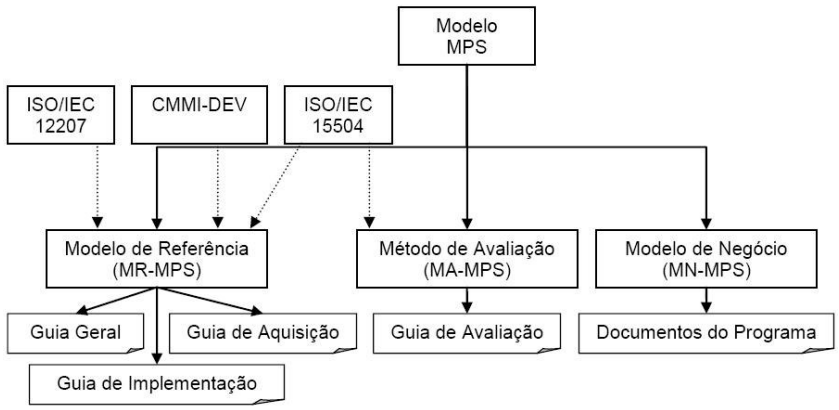


Figura 3.4. Estrutura do Modelo MPS.BR [SOFTEX 2009]

organizacional na execução dos seus processos de software. Este Método de Avaliação será detalhado na seção 3.4.2.

Conforme mencionado anteriormente, assim como no CMMI, o modelo MPS define uma escala de níveis de maturidade para que uma organização esteja em conformidade com um destes níveis, deverá realizar um grupo de processos específicos (indicam onde a organização deve focar para obter a melhoria pretendida) do nível corrente, 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, descrevendo o objetivo geral a ser atingido durante a execução do processo e os resultados esperados, que estabelecem os resultados a serem obtidos com a efetiva implementação do processo.

Por sua vez, cada processo possui um conjunto de atributos que correspondem à sua capacidade. Indica o grau de refinamento e institucionalização com que o processo é executado na organização. Busca 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, até definir 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.

Comment [W268]: Está meio confuso aqui.

Comment [W269]: Confuso. Os processos descrevem o objetivo geral a ser atingido durante a execução do processo?

Comment [W270]: Quem indica? Faltou elementos de coesão.

Comment [W271]: Quem Busca? Falta de coesão.

a) **Avaliação e Melhoria do Processo Organizacional.** Determinar o quanto os

melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos.

O nível B de modelo MPS.BR é composto pelos processos dos níveis de maturidade anteriores (G ao C) e não possui processos específicos. Contido, neste nível o processo de Gerência de Projetos sofre mais uma evolução, sendo acrescentados novos resultados para atender aos objetivos de gerenciamento quantitativo.

Comment [W272]: do

Comment [W273]: Contido?

O nível A de modelo MPS é composto pelos processos dos níveis de maturidade anteriores (G ao B), atendendo integralmente todos os atributos de processos e não possui processos específicos.

Comment [W274]: do

O processo de avaliação descreve o conjunto de atividades e tarefas a serem realizadas para atingir este propósito e possui quatro subprocessos, conforme ilustrado na figura Figura 3.6 a seguir, e obtém os seguintes resultados:

Comment [W275]: Este propósito? Que propósito? Melhorar isso.

O patrocinador pode ser um representante da alta gerência da unidade

unidade organizacional por uma terceira parte para fins de contrato. Para que uma avaliação seja conduzida com sucesso, é necessário:

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

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.

- O(s) avaliador(es) adjunto(s) da IA[8];

Tabela 3.4. Subprocesso Contratar a Avaliação [SOFTEX 2009]

Tabela 3.6. Subprocessos Realizar a Avaliação Final [SOFTEX 2009]

Tabela 3.7. Subprocesso Documentar os Resultados da Avaliação [SOFTEX 2009]

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, podemos concluir que poderá existir a situação de uma organização estar em conformidade com o MPS.BR, mas não com o CMMI; o que não acontecia na versão anterior, onde prevalecia a equivalência entre ambos.

Comment [W276]: Obstante? Nem sei o que significa. Hehe!

Comment [F277]: Ta ok =D

2. No CMMI com representação em estágios, qual nível de maturidade, o desempenho dos processos é controlado usando estatísticas e outras técnicas quantitativas?

Comment [W278]: Em qual nível...

Comment [W279]: Virgule desnecessária.

4. Quais são as áreas de processo que as empresas que estão no nível de maturidade 5

Comment [F280]: Dois "ques". Esse termo poderia ser removido.

6. Quais os pontos positivos e negativos do modelo CMMI?

Comment [W281]: Boa pergunta, mas a resposta está clara no texto?

O mercado de software tem evoluído exponencialmente juntamente com a popularização dos computadores e dispositivos móveis, fato este, que deriva da globalização e da necessidade de um mercado mais competitivo, 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 garantem, 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 de qualidade ou maturidade que acompanham ou propõem práticas e processos para produção de software e redução dos defeitos.

Portanto é importante notar que a infra-estrutura 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 infra-estrutura traz a uma iniciativa de MPS, a compreensão das suas funções e responsabilidades, não pode ser subestimado.

Este capítulo trata de 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 15504 e propõe uma engenharia de processo dirigida por perfis de capacidade de processo.

Comment [HV282]: ***"Adjunto"*** poderia ficar melhor?

Comment [HV283]: Palavra repetida sugere redundância?

Comment [HV284]: Muitas quebras de texto

Comment [HV285]: "Garantem" não possui concordância com "qualidade"

Comment [HV286]: Palavra repetida sugere redundância?

Comment [HV287]: Hífen foi extinto pela nova gramática

Comment [HV288]: Não ficaria melhor: aborda os

Comment [HV289]: ISO/IEC 15504

A primeira abordagem sistêmica de processos, data da década de 30 com Walter Shewhart (Shewhart, 1980), ele iniciou um trabalho em melhoria de processos com ênfase nos princípios do controle estatístico. Esses princípios foram refinados posteriormente na década de 80 em trabalhos de W. Edwards Deming (1986) e Joseph Juran (1997). Entretanto, estes trabalhos eram focados na indústria de manufatura.

Comment [HV290]: Texto confuso

Comment [HV291]: Cadê a conclusão da idéia iniciada depois do ponto?

Em diversas fontes da literatura direcionadas a processos de software, Moreira encontrou definições para o processo de software, mas em sua totalidade todas tem algo em comum. Veja abaixo:

Comment [HV292]: Não precisa dizer o ano ? ou pelo menos informar: Ainda segundo o autor citado acima?

Comment [HV293]: É viável excluir este trecho

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.

Comment [HV294]: Excluir a vírgula, não se separa substantivo de verbo

Estudos e pesquisas tem concentrado a Engenharia de Software em uma sub-área específica nomeada de Melhoria de Processo de Software (*Software Process Improvement - SPI*) ela 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].

Comment [HV295]: Hífen excluído pela nova gramática

Comment [HV296]: "denominada"

Comment [HV297]: Seria viável um ponto final, pois está havendo confusão e incoerência de idéias

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*.

Comment [HV298]: Retirar vírgula

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.

Comment [HV299]: Retirar vírgula

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.

Comment [HV300]: Seria viável um ponto final, pois está havendo confusão e incoerência de idéias

Comment [HV301]: Retirar a vírgula

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.

Comment [HV302]: Retirar a vírgula

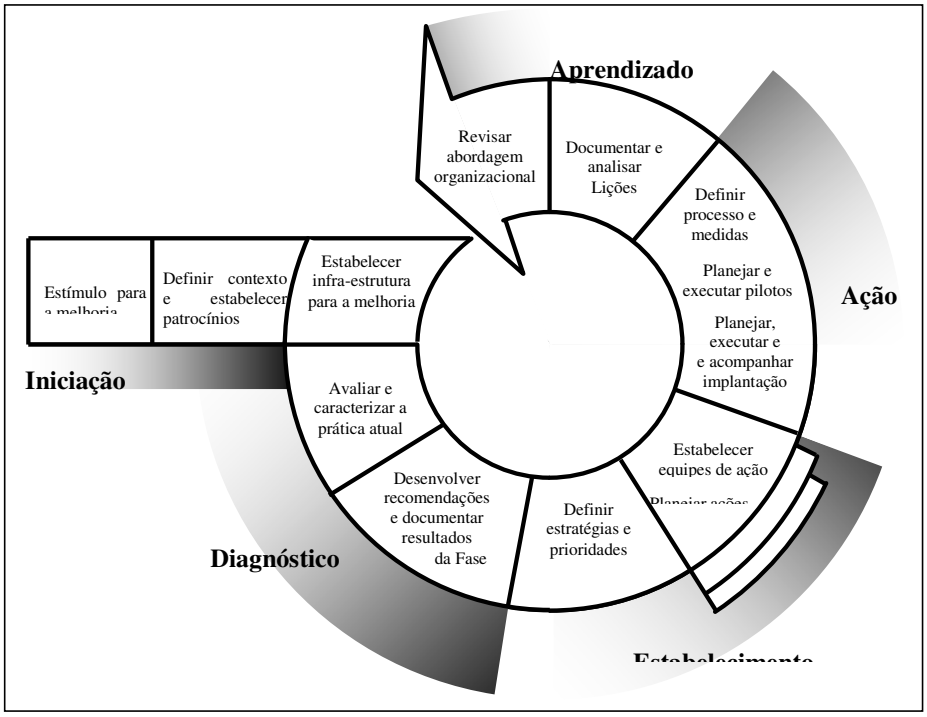


Figura 3 - Tradução do Ciclo do modelo IDEAL, Fonte: Moreira (2008) Apud McFeeley, 1996

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 McFeleey, para a implementação do IDEAL:

Comment [HV303]: Precisar dizer isso?

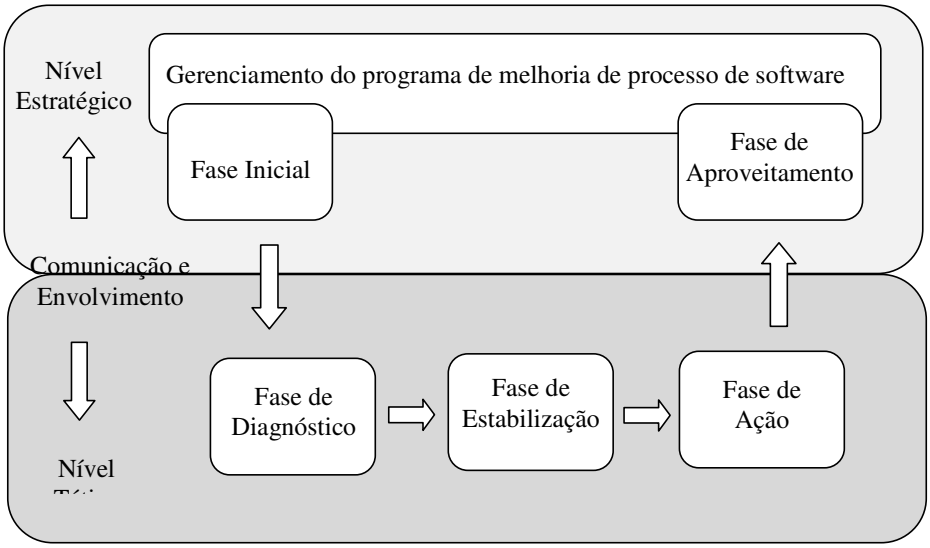


Figura 4 - Duas dimensões da atividade de melhoria de processos através do Ideal, Adaptado de McFeelev. 2006.

Dito isto, vamos à próxima seção onde serão explicadas detalhadamente, 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.

Comment [HV304]: Dito o que? O leitor fica vago se você não especificar a idéia. "***Ist"o*** se refere ao parágrafo em que você esta abordando e não aos parágrafos lidos anteriormente.

Comment [HV305]: Não é para ser impessoal?

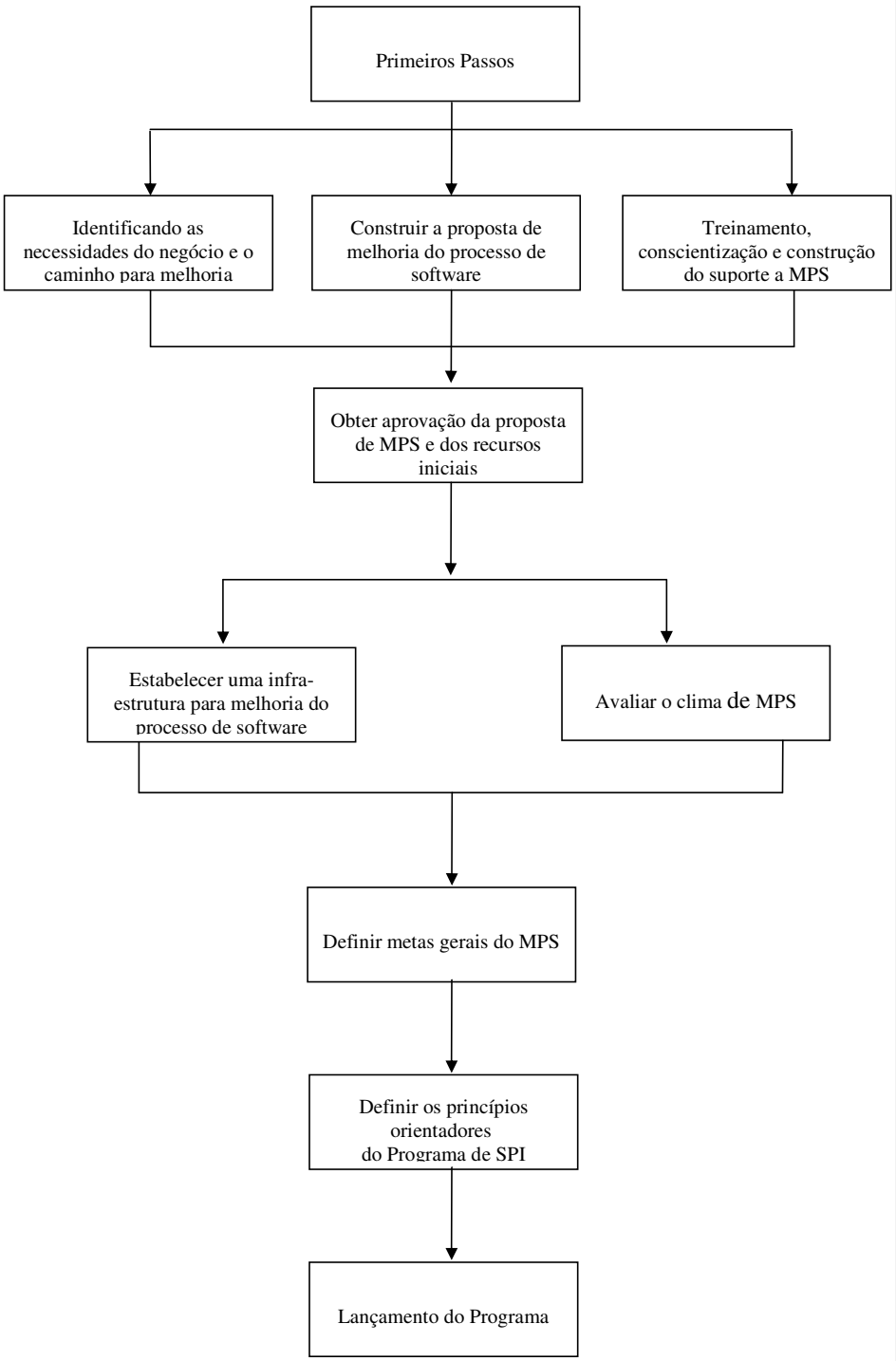
Assim também acontece com o IDEAL, o modelo é composto por cinco fases e uma atividade Gerencial do Programa de Melhoria de Processo de Software, as primeiras são formadas de atividades-processos que devem ser seguidas para um maior alcance dos objetivos organizacionais, a segunda é essencial para o controle e evolução da MPS, aquelas e esta serão explicadas nas seis seções que procedem.

Comment [HV306]: Quebra de idéias. É necessário um ponto "."

Comment [HV307]: Confusão de idéias e texto sem coerência

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.

Comment [HV308]: Referências?


Primeiros Passos
Identificando as necessidades do negócio e o caminho para melhoria
Construir a proposta de melhoria do processo de software
Treinamento, conscientização e construção do suporte a MPS
Obter aprovação da proposta de MPS e dos recursos iniciais
Estabelecer uma infra-estrutura para melhoria do processo de software
Avaliar o clima de MPS
Definir metas gerais do MPS
Definir os princípios orientadores do Programa de SPI
Lançamento do Programa

O grupo de gerenciamento (*MSG*) deve compreender a base da organização e do processo atual de software, para que, ele possa desenvolver um plano que permita atingir o negócio na mudança específica no 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.

O resultado principal desta fase são as conclusões finais e relatório de recomendações, que é produzido como resultado das atividades de *baseline*. Saídas secundárias podem ser revisões à visão da organização e do plano de negócios, um conjunto mínimo recomendado de *baselines* inclui:

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 o pessoal da própria organização podem ser treinados 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

Comment [HV309]: para que

Comment [HV310]: Deste o que ? Incoerência de ideias com o parágrafo anterior

Comment [HV311]: O que são atividades de *baseline*? Não seria viável uma nota de rodapé?

Comment [HV312]: Retirar vírgula

Comment [HV313]: O que é pessoal? Concordância errada com “treinados”

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.

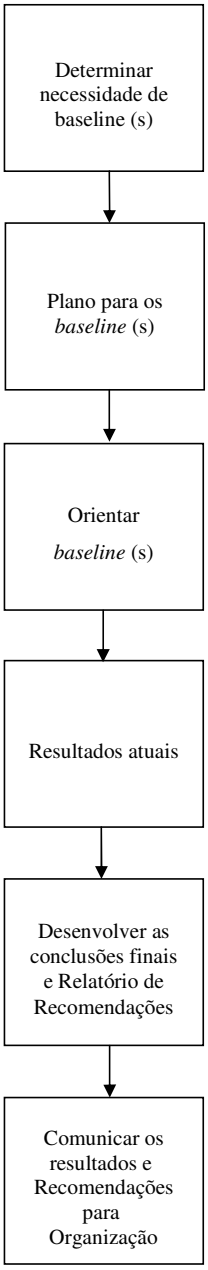


Figura 6 - Fluxo do Processo da fase de Diagnóstico, Adaptado de [McFeeley 2006]

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, juntamente com os resultados do *baselining* de esforços.

Esta é uma fase que 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, tendo 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).

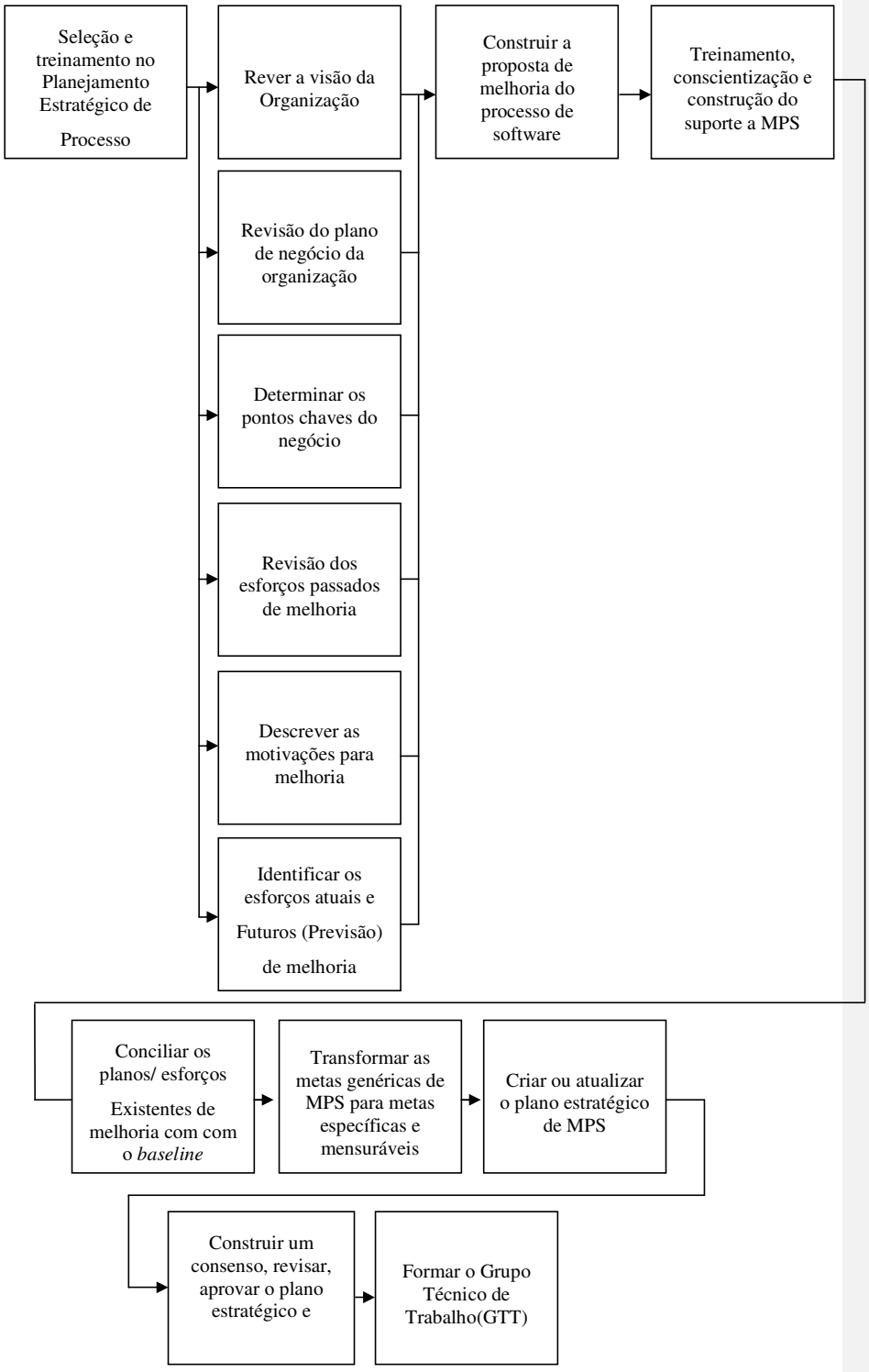
Além da produção de um plano estratégico de MPS é preciso integrá-lo com iniciativas já previstas ou em andamento de Gestão da Qualidade Total (*TQM*), com as conclusões e recomendações da *baseline* no plano de ação estratégico e alinhá-lo com o plano de negócio da organização, missão e visão. A Figura 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

Comment [HV314]: Retirar vírgula

Comment [HV315]: adjuntos aos

Comment [HV316]: Que frase?

Comment [HV317]: Coloquial


Seleção e treinamento no Planejamento Estratégico de Processo
Rever a visão da Organização
Construir a proposta de melhoria do processo de software
Treinamento, conscientização e construção do suporte a MPS
Revisão do plano de negócio da organização
Determinar os pontos chaves do negócio
Revisão dos esforços passados de melhoria
Descrever as motivações para melhoria
Identificar os esforços atuais e Futuros (Previsão) de melhoria
Conciliar os planos/ esforços Existentes de melhoria com com o *baseline*
Transformar as metas genéricas de MPS para metas específicas e mensuráveis
Criar ou atualizar o plano estratégico de MPS
Construir um consenso, revisar, aprovar o plano estratégico e
Formar o Grupo Técnico de Trabalho(GTT)

A fase de ação é a fase onde as melhorias são desenvolvidas, postas em prática e implantadas em toda a organização. As melhorias que vários grupos envolvidos no trabalho de melhoria proporam, 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.

Comment [HV318]: a que provém

Comment [HV319]: Retirar a vírgula

Comment [HV320]: Como assim? Incoerência com a idéia citada na primeira linha. Confusão para o leitor.

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 a práticas atuais em "como estão". E assim determinar a áreas de melhoria, e como os processos candidatos a melhoria devem ser examinados e avaliados.

Comment [HV321]: Colocar um ponto final "."

Comment [HV322]: Colocar vírgula

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.

Comment [HV323]: Ponto e vírgula?

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

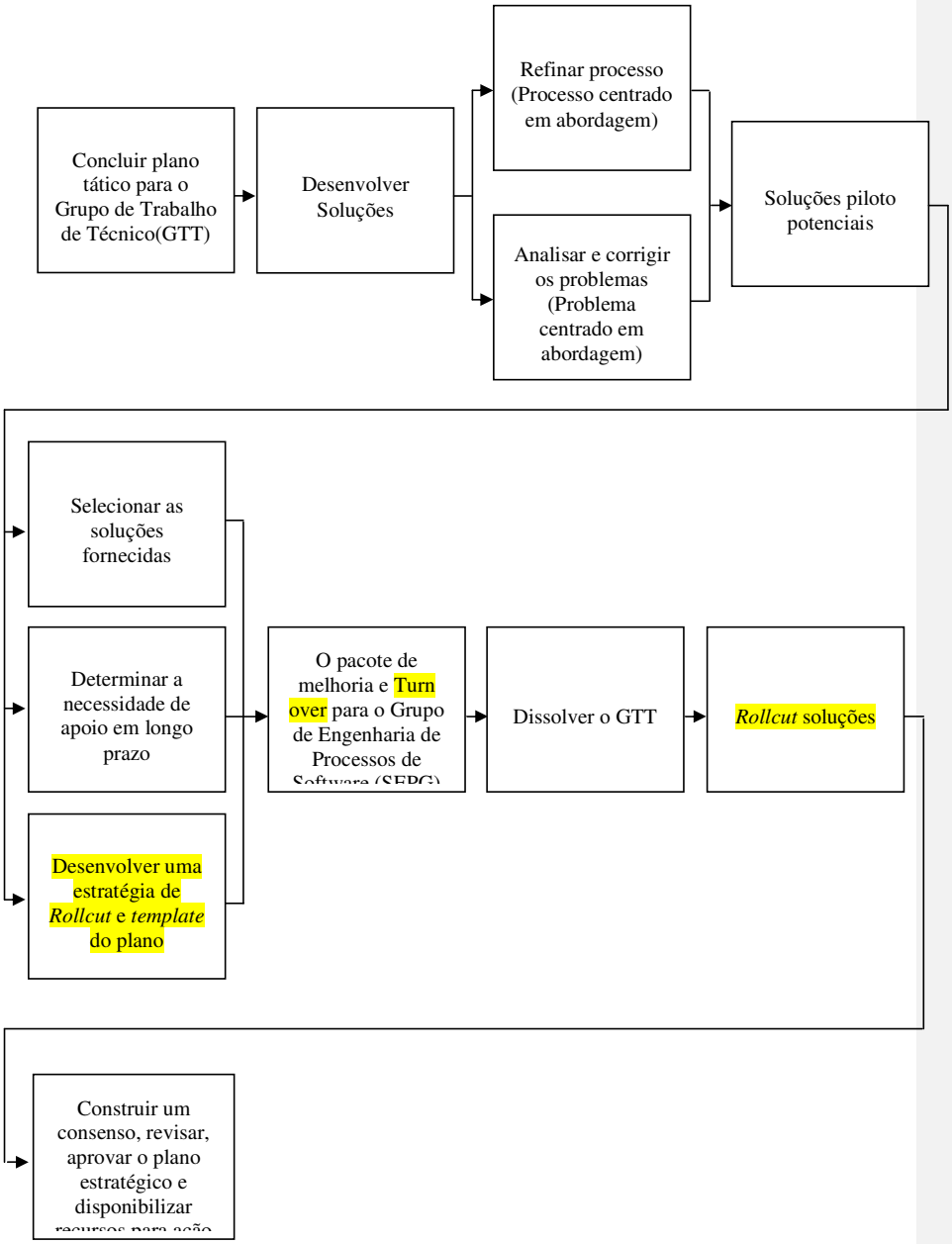


Figura 8 - Fluxo do processo da fase de ação, Adaptado de [McFeeley2006]

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 reentrar no IDEAL na fase de diagnóstico. Aproveitando esta fase, além de preparação para o próximo ciclo através IDEAL, dá-lhe a oportunidade de ajuste a melhoria do processo de software (MPS) antes de iniciar o processo novamente.

Comment [HV324]: ?

Comment [HV325]: Dar-lhe

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:

Comment [HV326]: ?

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 dada início a caracterização da melhoria contínua, baseada nas mudanças de melhoria dos processos e nas lições aprendidas com a MPS.

Comment [HV327]: É dado o início da

A Figura 7 ilustra os sete processos sugeridos pelo guia de implantação do IDEAL para a fase de aproveitamento, após esta, seguiremos com a fase de gerenciamento do programa de melhoria do processo de software da organização para dar continuidade a MPS.

Comment [HV328]: Impessoal?


Determinar necessidade de baseline (s)
Analisar as lições
Revisar a abordagem organizacional
Revisão dos patrocinadores e compromisso
Estabilizar metas de alto nível
Desenvolver nova revisão da proposta de melhoria do processo de software (MPS)
Continuar com a MPS

A melhoria do processo de software é um empreendimento 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.

Comment [HV329]: ?

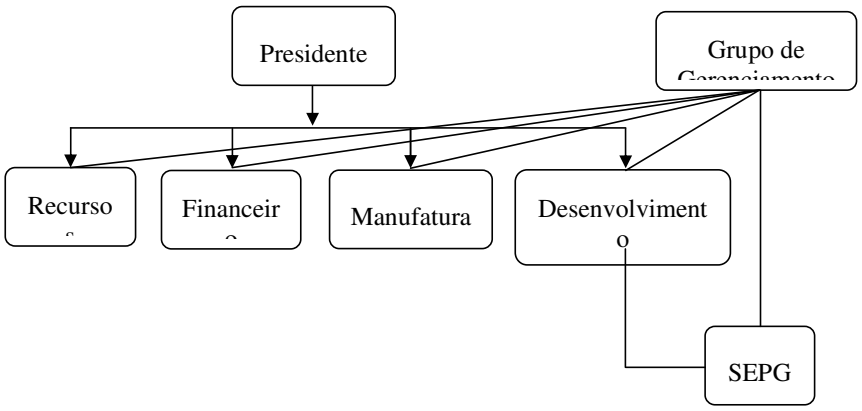


Figura 10 – Componentes típicos de uma infra-estrutura de MPS , Adaptado de [McFeeley2006

As atividades de melhoria não irão ocorrer em um vácuo nem ocorrerão em série. Depois que o programa de MPS está em curso, haverão várias atividades de melhoria ocorrendo em diferentes unidades organizacionais. Por exemplo, pode 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.

Comment [HV330]: Como exemplo, pode haver

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.

O PRO2PI[9], 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.

Exemplos de modelos mais utilizados são a Norma ISO/IEC 12207 (1998[10]), 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,

[9] 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).

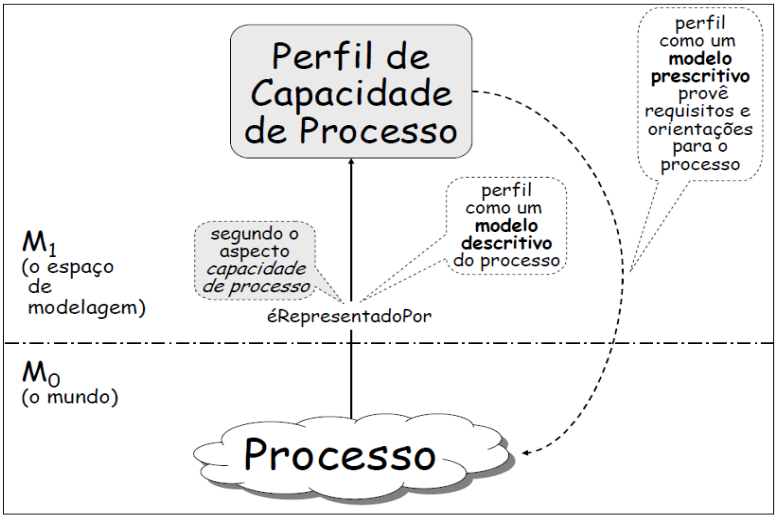


Figura 11 - Perfil de capacidade de processo e processo, Fonte: (Salviano 2006)

De acordo com Salviano 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 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.

As propriedades do PRO2PI foram definidas a partir de seis fases, onde foram

Comment [HV331]: Citação direta, indrireta ou de autoria própria? Acima de 3 linhas!!!

Comment [HV332]: ?

Comment [HV333]: Palavras em inglês devem ter formatação Itálica

resumo das fases de formação de um PRO2PI está ilustrado na figura 4 e podemos chamá-lo de "processo de montagem do PRO2PI".

Comment [HV334]: Figura 4

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.

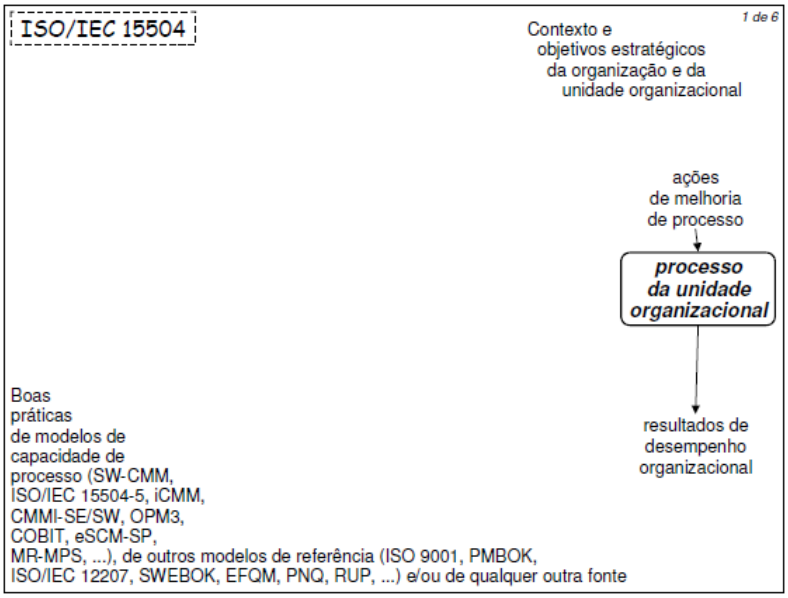


Figura 12 - Diagrama da utilização corrente da melhoria de processo, Fonte (Salviano, 2006).

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

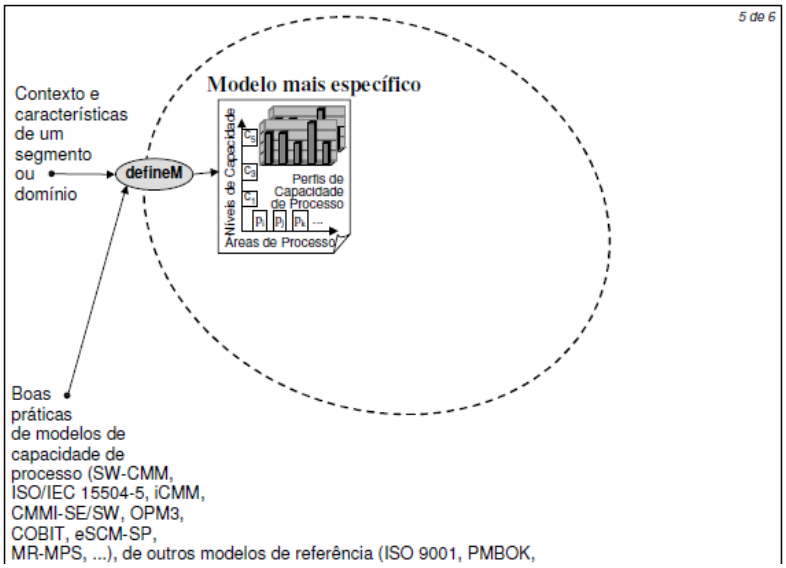
utilização da versão corrente do perfil. Esse uso é representado pela função “defineP” (define, ou atualiza, perfil de capacidade de processo) da Figura 3, (Salviano, 2006).

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].

Comment [HV335]: Parênteses ou colchetes?

A figura que ilustra 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.

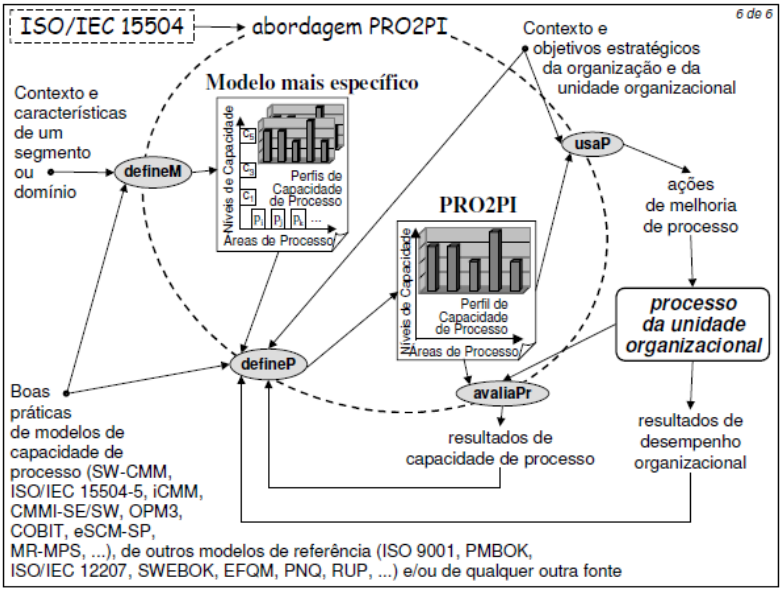
Comment [HV336]: Excluir



A seleção de referencias é feita e "*esse modelo mais específico pode conter boas práticas selecionadas dos modelos de capacidade de processo, modelos de outros tipos ou de qualquer outra fonte de forma a representar orientações relevantes no segmento ou domínio. Esse uso é representado pela função "defineM" (define, ou atualiza, modelo mais especifico de capacidade de processo*" [Salviano 2006] da Figura 3.

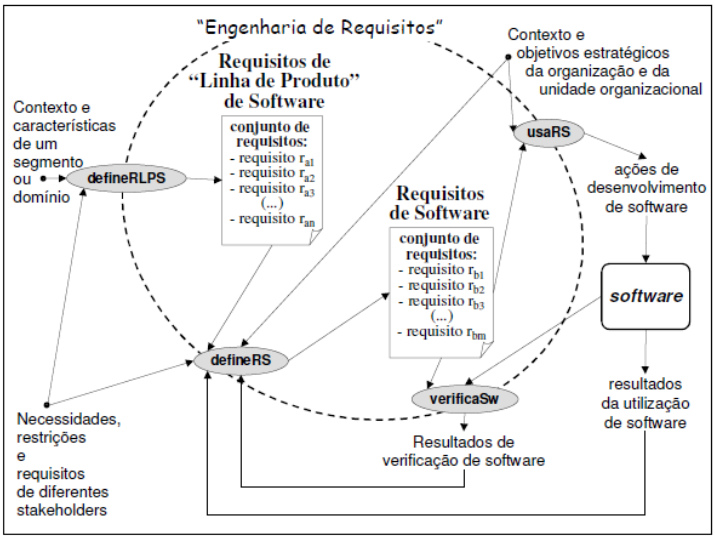
Comment [HV337]: Referência ou citação direta? Informar página!

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.



A Figura 5 mostra 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.

Comment [HV338]: demonstra



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.

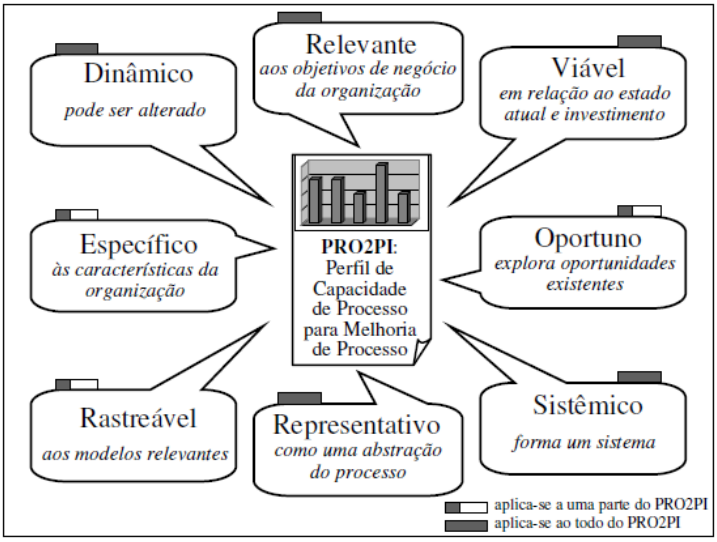


Figura 8 - Propriedades de PRO2PI, Fonte: (Salviano, 2006)

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*).

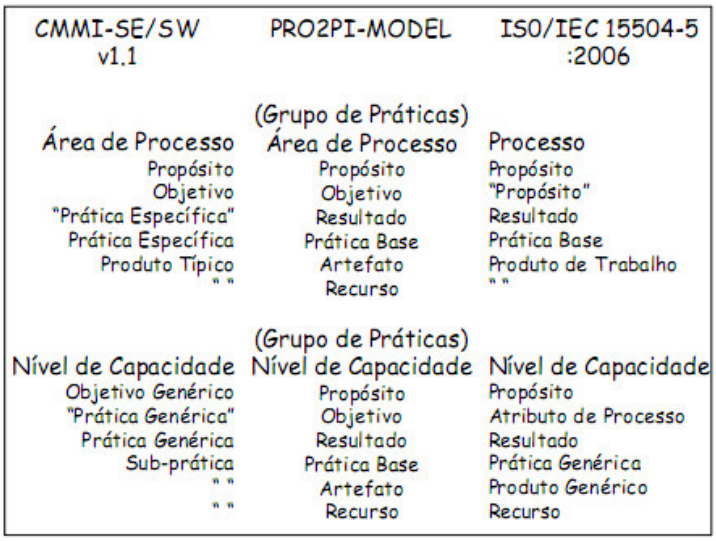
Comment [HV339]: Tabela 1

	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

	definida em PRO2PI-MEAS.

- Buscar representar em um PRO2PI os elementos de outros modelos de

SMMM, KMMM, PMMM, UMM, TMM, OOSPICE, SPICE4SPACE, Automotive SPICE, 15504MPE e S9K;



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

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.

Comment [HV340]: ?

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

Comment [HV341]: Retirar a vírgula

Comment [HV342]: Viabilidade = 1

Comment [HV343]: Viabilidade > 1

Comment [HV344]: Viabilidade<1

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.

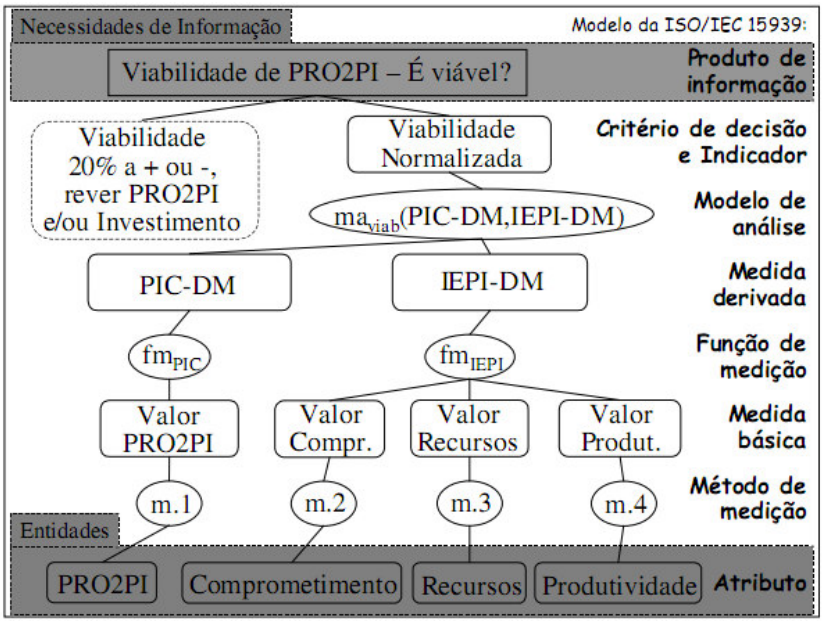


Figura 17 - Produto de informação Viabilidade de PRO2PI, Fonte: (Salviano, 2006)

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

porque o objetivo é a melhoria do negócio da organização, por meio da melhoria do processo de produção software.

Assim finalizamos 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.

Comment [HV346]: Impessoal?

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. Wang (2008) define o Seis Sigma como uma abordagem que melhora a qualidade

através da análise de dados estatísticos. Nos últimos anos tem havido um aumento significativo em sua utilização.

"*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 [HV347]: Citação direta? Cadê a página? Acima de três linhas

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 [HV348]: Citação direta ou indireta?

Six Sigma possibilita uma capacidade de processo que deve gerar apenas 3,4 defeitos por milhão de oportunidades (DPMO), apresentando 99,99966% de perfeição. 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.

Comment [HV349]: Quem disse isso?

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.

Comment [HV350]: Parênteses ou colchetes?

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.

Comment [HV351]: O

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 pela que foram

baseadas a definição da abordagem IDEAL [McFeeley 1996] e o ciclo de melhoria da ISO/IEC 15504.

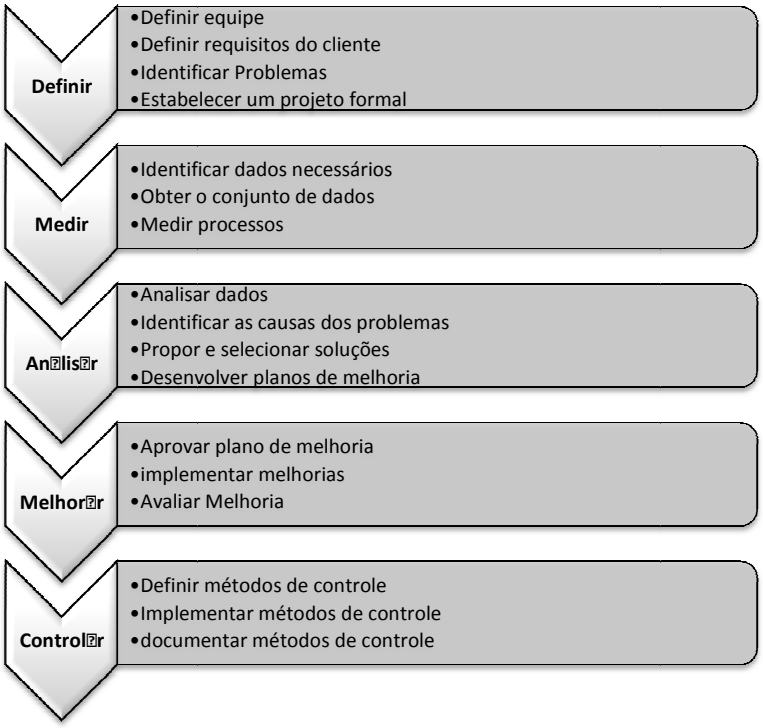
Inicialmente o ciclo PDCA foi idealizado por Walter Shewart na década de 30, mas o mesmo só viria a ser disseminado largamente na indústria por Deming [1986], o nome do modelo PDCA é derivado do acrônimo das palavras (*Plan-Do-Check-Action,* Planejamento-Execução-Verificação-Ação) o modelo é visto na literatura como o precursor no conceito de melhoria de processos [Moreira 2008, Salviano 2006].

Comment [HV352]: Um ponto final "."

Comment [HV353]: Quebra de ideias

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.

Comment [HV354]: Quem disse isso?



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 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.

Comment [HV355]: Itálico

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.

➔Muitas palavras repetidas em seqüência. É viável o uso de um dicionário para buscar sinônimos mais suaves.

Seção 3 –Inspeção de Software: Nesta seção, serão apresentadas as etapas da inspeção de software e as ferramentas de apoio à inspeção.

Seção 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.

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 diferente aspecto 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 a serem respeitados.

Quando um produto é medido em relação a uma série de características, é

Comment [j356]: Só esta tem nesta seção, os outros é serão e são abordados

Comment [j357]: Verificar se está no padrão dos outros, achei resumido d+

Comment [j358]: Após a criação [não seria melhor?]

Comment [j359]: Reescrever, descrevendo quais os padrões e qual a sua utilidade.

Comment [j360]: Parágrafo muito pequeno, unir com o proximo ou colocar no início da introdução junto com o primeiro parágrafo.

Comment [j361]: Para cada aspecto diferente [fica melhor]

Comment [j362]: E qualitativos não?

podem ser usadas para uma definição mais precisa de requisitos, fixando-se no início do projeto, os valores desejados para o produto final.

1.1. 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.

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:

Engenharia de Software – Qualidade do Produto -- Parte 1: Modelo de Qualidade

Esta norma pode ser aplicada nos seguintes momentos:

- Avaliação da especificação de software para verificar se ele irá satisfazer aos requisitos de qualidade durante o desenvolvimento;

Comment [j363]: Acho que ficaria melhor **normas de qualidade de produto**

Comment [j364]: As normais, x,y,z,v que serão explicadas com mais detalhes nas seções a seguir.

Comment [j365]: Deletar espaço após o em

Comment [j366]: Mudar simbolo para -

Comment [j367]: Nas estapas seguintes

Comment [j368]: Apagar espaço após satisfazer

1.1.1.2. Características e sub-características de qualidade de software

Comment [j369]: subcaracteristica

Comment [j370]: A maior parte dos modelos de qualidade

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.

Comment [j371]: Falta de referência

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].

Comment [j372]: Por haver tópicos relacionados abaixo, é melhor começar o parágrafo assim: Segundo [fulano].... abaixo:

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.

Comment [j373]: Colocar marcadores e tirar italic e substituir por negrito se necessário

Tabela 1.1. Características de qualidade interna e externa (NBR ISSO/IEC 9126-1, 2001)

	relacionamento entre nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas.
	Conjunto de atributos que evidencia o esforço necessário para fazer modificações necessárias no software.

Comment [j374]: Justificar, toda a tabela e centralizar as colunas, e ter cuidado para tabela não ficar divida em paginas diferentes

Caracteristicas	Subcaracteristicas
Funcionalidade (satisfação das necessidades)	• Adequação (execução do que é apropriado) • Acurácia (execução de forma correta) • Interoperabilidade (interação com quem deve) • Conformidade (aderência às normas) • Segurança de acesso (bloqueio de uso não autorizado)
Confiabilidade (imunidade a falhas)	• Maturidade (freqüência das falhas) • Tolerância a falhas (forma de reação à falhas) • Recuperabilidade (forma de recuperação de 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 (recursos utilizados)
Manutenibilidade (facilidade de manutenção)	• Analisabilidade (facilidade de encontrar falha) • Modificabilidade (facilidade de modificar) • Estabilidade (baixo risco quando 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 [j375]: Fazer tabela manualmente, em p&b

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, considerando estes requisitos. A norma divide-se em itens apresentados na Tabela 1.2.

Tabela 1.2. Itens da Norma 12119

Comment [j376]: referencia

Comment [j377]: Pode ocultar isso

Comment [j378]: Esta norma

Comment [j379]: Não entendi essa tabela, referencia faltando

A norma inclui detalhes que devem estar presentes no produto, visando também facilitar o trabalho do avaliador, tais como:

Comment [j380]: Esta norma

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 1.3:

Tabela 1.3. Tipos de certificação

Certificação		

Norma		
		Ensina a utilizar as outras normas do grupo. Define os termos técnicos utilizados nas demais partes, contém requisitos gerais para especificação e avaliação de qualidade de software e esclarece os conceitos gerais.

Comment [j381]: Apagar espaço depois de software

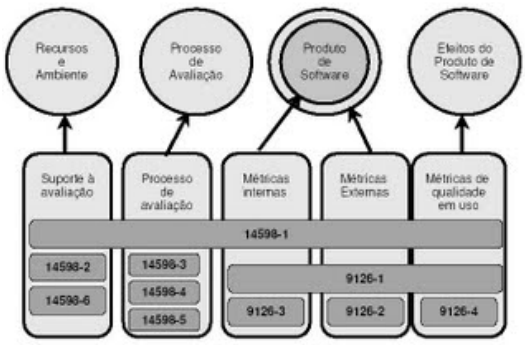
Comment [j382]: Apagar espaco

Comment [j383]: Centralizar colunas de todas as tabelas, e colocar referencia em todas

Comment [j384]: Tirar espaço

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. Junto com as normas NBR ISO/IEC 14598-2 e NBR ISO/IEC 14598-6 estabelecem itens necessários para o suporte à avaliação e junto com 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 1.2.

Comment [j385]: Ã?



Comment [j386]: Figura ilegível

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 1.5 - um modelo de relatório de avaliação, segundo um anexo da norma 14598-5.

Comment [j387]: Apagar espaco

Seção	
	Identificação do avaliador

	Identificação do contratante e fornecedor

1.1.4. Projeto SQuaRE

Comment [j388]: 1.1. É um Modelo de qualidade de produto, ou um manual?

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.

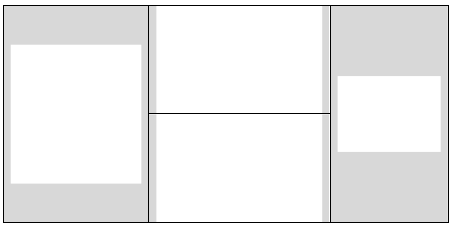
Comment [j389]: Referencia?

1.1.5. Norma SQuaRE

Comment [j390]: É uma norma mesmo?

Na reorganização das antigas normas 9126 e 14598, o projeto SQuaRE adotou uma divisão de assuntos em cinco tópicos que aparecem na Figura 1.3:

Comment [j391]: ilustrados



Cada divisão é composta por um conjunto de documentos e trata de um assunto em particular, como apresentadas abaixo:

- Gerenciamento: os documentos desta divisão da norma 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 um dos atores envolvidos espera de um produto.

- 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, evidentemente, da especificação dos requisitos do software.

Vale enfatizar que o projeto SQuaRE não surgiu para desmentir as normas 9126 e 14598, mas sim para organizá-las.

Comment [j392]: Que tratam de assuntos específicos, e referenciar, os tópicos q seguem aqui!

Comment [j393]: Apagar espaço e colocar negrito

Comment [j394]: Ã? reescrever

Comment [j395]: Do que cada autor espera...

Comment [j396]: Afronta ao leitor

Comment [j397]: Usar outra palavra

Segundo Sommerville [Sommerville 2004], dentro do processo de V & V existem duas abordagens para checar e analisar um sistema: Teste de Software e Inspeção de Software. 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.

Comment [j398]: Não repetir o nome do autor, deixar so data dentro do []

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, a menos que seja um programa bem pequeno e simples. 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.

Comment [j399]: Mesmo assim , poderao haver erros , reescrever paragrafo

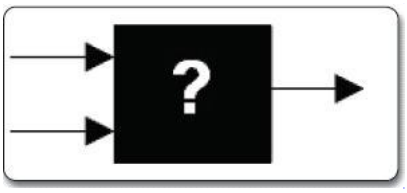
Comment [j400]: Não concordo, defeitos é algo grave, não seria erro?

Para adequar 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 1.4 serão apresentados os 3 principais modelos. Nas próximas sub-seções serão apresentados conceitos fundamentais inerentes ao processo de teste de

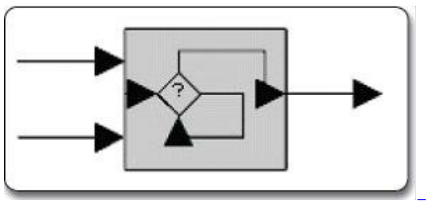
Comment [j401]: Reescrever, falta algum

1.2.1. Abordagens de Testes

Comment [j403]: Topico sem referencia



Comment [j404]: Já vi essa figura em algum lugar, qual a referência?



Comment [j405]: Referência?

1.2.2. Estágios de Testes

Comment [j406]: Tópico sem nenhuma referência

- 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 testes 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 *debbuging*

Comment [j407]: Diminuir a repetição da palavra testes

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.

- 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 achar defeitos o mais cedo possível 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 é testada a integração dos componentes do software entre si, mas também destes 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 faz aquilo que é suposto fazer. 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.

1.2.3. Tipos de Testes

Cada tipo de teste é focado em um grupo de atividades com um determinado

Comment [j408]: Colocar uma linguagem mais formal

Comment [j409]: Testada? estranho

Comment [j410]: Linguagem mais formal

Comment [j411]: Tópico sem referencia

Comment [j412]: ocultar

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.

Segundo [Graham et. al 2007], o processo de testes pode ser dividido basicamente em cinco etapas: *planejamento e controle, análise e projeto,*

Comment [j413]: acho que teria aspas aqui se

projeto específico, podem se sobrepor, serem executadas em paralelo ou até mesmo serem repetidas.

1.2.4.1. Planejamento e Controle

Comment [j415]: referencia

1.2.4.2. Análise e Projeto

Comment [j416]: referencia

1.2.4.3. Implementação e Execução

Comment [j417]: referencia

- Criar um *log* com as saídas da execução dos testes e registrar os identificadores e versões do software que está sendo testado.

Comment [j418]: tirar espaco

1.2.4.4. Avaliação do critério de saída e relatório

Comment [j419]: referencia

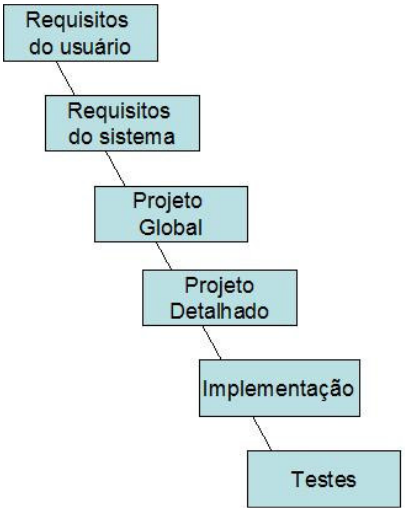
1.2.4.5. Atividades de encerramento de teste

Comment [j420]: referencia

1.2.5. Testes ao longo do ciclo de vida de Software

Comment [j421]: referencia

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 contrario, não conseguirá obter o sucesso desejado.

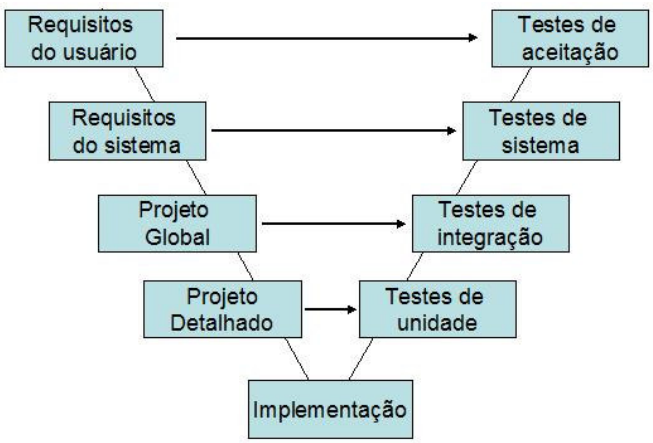


Comment [j422]: pode figura colorida?

Basicamente, o ciclo de desenvolvimento do modelo usa os quatro estágios de teste explicados neste capítulo: testes unitários, testes de integração, testes de sistema e testes de aceitação. Cada estágio pode ser ajustado com as etapas do clico de desenvolvimento de 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 1.7.

Comment [j423]: referencia

Comment [j424]: pode figura colorida



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 tarde, mais difícil de se 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, [Fagan 1986] a utilização de inspeções informais de software capturam em torno de 60% dos erros em um programa. Mills et al. [Mills et al. 1987] sugere que uma aplicação mais formal de inspeção de software pode detectar até mais de 90% dos erros de um programa. Selby e Basili [Selby et al. 1987] 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.

[Boehm e Basili 2001] ressaltam ainda que inspeções e testes capturam diferentes tipos de defeito e em diferentes momentos do processo de desenvolvimento de software. 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 inicias do processo de desenvolvimento do software, e uma vez que o sistema esteja integrado, é necessária a introdução de testes para checar as propriedades do sistema e ver se este está de acordo com o desejo do cliente.

Comment [j425]: reescrever a idéia

Comment [j426]: ocultar referencia fora do colchetes ja que tem et al.

Comment [j427]: 71

Comment [j428]: Falta virgule entre a data

Comment [j429]: iniciais

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. Uma boa variedade de inspetores, representando diferentes áreas de conhecimento, é interessante para o processo de inspeção. O papel de cada participante será explicado abaixo.

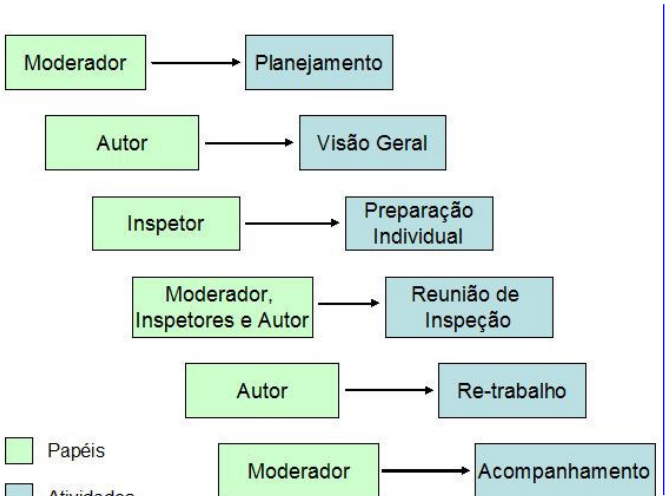
Comment [j430]: melhor se em outro paragrafo, e reescrever ideia

- 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.

Comment [j431]: colocar negrito no incio dos tópicos?

- Inspetor – examina o produto antes da reunião de inspeção (fase de preparação) e durante de modo a tentar encontrar defeitos. Pode também identificar problemas amplos que estão fora do escopo da equipe de inspeção, como também sugerir melhorias.

Comment [j432]: reescrever num p entender o início



Comment [j433]: p & b

- 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 contrario, o material deverá ser dividido em tamanhos menores, com inspeções a serem realizadas para cada uma destas partes.

- 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 contrario, ele pode remarcar a reunião.

- 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, todos 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 um tempo na etapa de re-trabalho.

Comment [j434]: seguir padrao dos outros [ou – ou : após tópico], e colocar negrito no tópico

Comment [j435]: seria um ponto? Falta acento

Comment [j436]: acento

Comment [j437]: exlcuir espaço

Comment [j438]: ocultar

Comment [j439]: mudar expressão

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 fim da década de 90 surgiram as ferramentas para Internet.

Comment [j440]: subtituit pronto por virgula

Comment [j441]: final

A seguir será apresentada uma ferramenta representante de cada geração introduzida anteriormente. 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.

Comment [j442]: [Hedberg 2004]? Eo que se segue?

- 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 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

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 dado. Um *browser C++* pode automaticamente apresentar itens relevantes de *checklist*[11] 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: por ser baseada em WEB, permite que os inspetores acessem a aplicação de seus próprios computadores; permite 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.

Foi desenvolvido porque a IQUIP (Intelligent & Quick Information Processing) e seus clientes que necessitavam de um modelo que suportasse as constantes mudanças do processo de testes. 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.

Comment [j444]: Dados?

Comment [j445]: Já disse acima, pode ocultar

Comment [j446]: Repetir menos
Esta palavra, pode oculta-la a frente

Comment [j447]: Reescrever ideia

O processo de teste é parte do processo total de desenvolvimento, demonstrado na Figura 1.9, por isso é de extrema importância analisar os problemas relacionados às atividades de teste e que impactos eles causam no processo geral [JACOBS & Sotokes 2000].

Comment [j448]: E os impactos que eles causam fica melhr?

Comment [j449]: Refazer figura traduzida

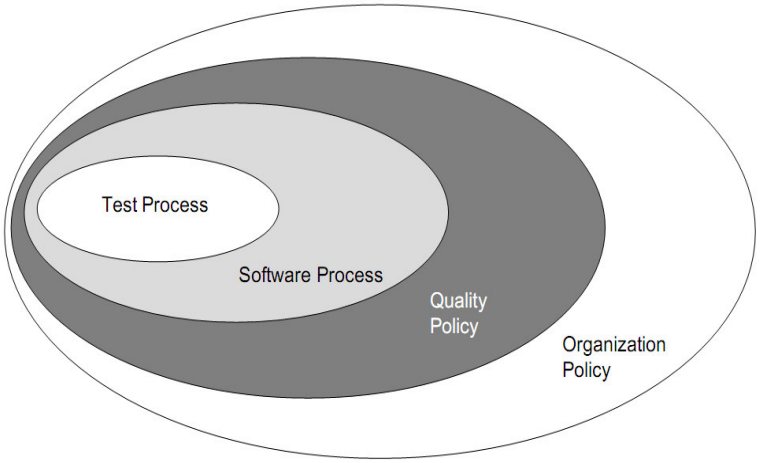
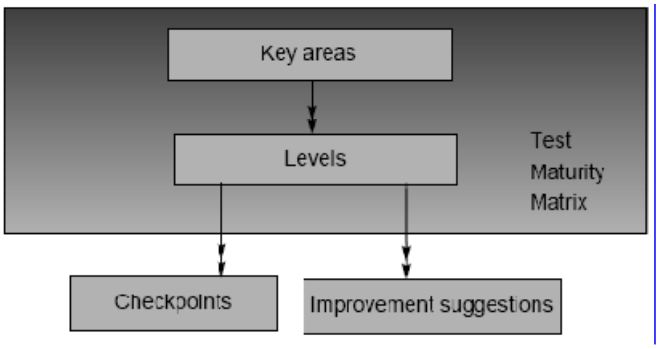


Figura 1.9. Processo Total de Desenvolvimento

Comment [j450]: Refrencia ou das autoras?



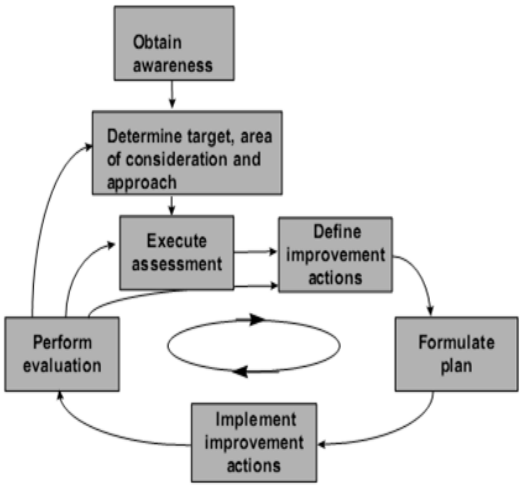
Comment [j451]: Refazer figura traduzida, referencia?

1.4.1.2. Áreas Chave

1. 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.

Comment [j452]: De que? referência

Comment [j453]: Padronizar [negrito : ou -]



Como pode ser observado na Figura 1.11, há uma série de passos, detalhados abaixo, para se implantar a melhoria do processo de testes. É preciso fazer uma análise de como está o processo atual, para verificar qual é o objetivo e estruturar a empresa para alcançá-lo [Koomen & Pol 1999]. Os passos são:

Comment [j454]: Referencia? Refazer traduzindo

Comment [j455]: Reescrever, de acordo com [Koomen & Pol 1999]. Os passos são.....

- 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.

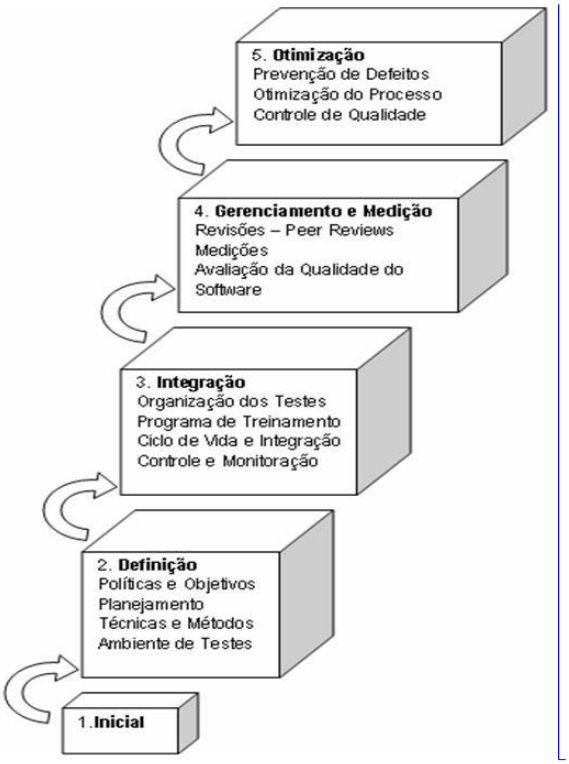
Comment [j456]: [Koomen & Pol 1999].?

O Testing Maturity Model – TMM [Burnstein, Homyen, Grom, & Carlson 1998] foi desenvolvido pelo Illinois Instituty of Tecnology 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 1.12 pode-se observar a estrutura do TMM.

Comment [j457]: Em itálico

Comment [j458]: et al. , data

Comment [j459]: apagar espaço



Comment [j460]: refazer para letras nao ficarem pixeladas, referencia?

1.4.2.1. Níveis de Maturidade do TMM

Comment [j461]: Dar uma introdução antes de falar dos tópicos, referência, acho melhor dividir falar mais e dividir em subseções os tópicos

- Nível 1 – Initial

Comment [j462]: traduzir

- Nível 2 – Phase Definition

Comment [j463]: traduzir

- Nível 3 – Integration

Comment [j464]: traduzir

Os testes são completamente integrados ao ciclo de vida do software, sendo

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.

- Nível 4 – Management and Measurement

Comment [j465]: traduzir

- Nível 5 – Optimization

Comment [j466]: traduzir

Os cinco de níveis de maturidade mostram uma evolução de um caótico e indefinido para um controlado e otimizado processo de testes.

Comment [j467]: reescrever

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.

Comment [j468]: [Koomen e Pol, 1999

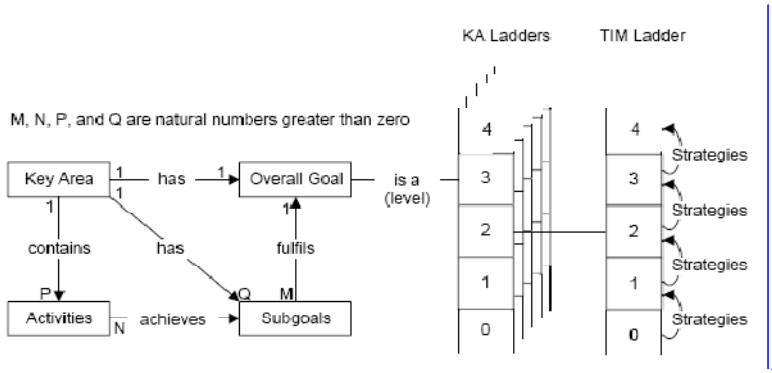
1.4.3.1. Modelo de Maturidade

Comment [j469]: referencia

Consiste em quatro níveis, o primeiro foi denominado de nível 0, considerado um nível de não conformidades e não será discutido neste livro, 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

Comment [j470]: Pq?

Na Figura 1.13 pode-se observar como o TIM está estruturado. Uma Área Chave contém várias atividades, deve-se executar uma série delas para alcançar um sub-objetivo. Uma Área Chave tem vários sub-objetivos e para atender um objetivo geral os subo-bjetivos devem ser atendidos primeiramente e os *checkpoints* devem ser identificados. Outro fator importante é que as Áreas Chaves possuem os mesmos nomes dos Níveis.



- Nível 1 – Baselining

Comment [j471]: corrigir

Comment [j472]: refazer e traduzir

Comment [j473]: padronizar tópicos, negrito. Etc.

1.4.3.2. 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:

- Aspecto: Organização

Comment [j474]: De que

Comment [j475]: Tocar por e

Comment [j476]: padronizar

- No nível *Optimizing,* atividades de planejamento e a rastreabilidade é continuamente melhorada baseada na análise de

métricas. Reuniões de *post-mortem*[12] são realizadas e os resultados armazenados e distribuídos.

[12]

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 esforço 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 a tempo, 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 atingir os objetivos desejados, buscando melhorias na qualidade do processo de teste de software, que afeta diretamente a qualidade do produto.

Comment [j477]: em que sentido, qualidade não é sinônimo de sucesso!

Comment [j478]: Ã?

Comment [j479]: Tem acento?

Comment [j480]: ocultar

Comment [j481]: quais? Pode ocultar, e continuar em melhorias

Exercícios

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.

Comment [j482]: ocultar espaço

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.

Comment [j483]: este livro é indicado como amterial de estudo p certificação ....

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

Comment [j484]: acho q nao cabe aqui

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.

Referências

Comment [j485]: padronizar todas

Uma introdução ao SWEBOK

Em uma conferência da OTAN em no ano de 1968, - *Software Engineering*: *Concepts and Techniques. Proceedings of the NATO Conferences* - Ronald Graham comentou "construímos sistemas como os irmãos Wright construíam aviões - constrói-se de uma só vez só, empurra-se para o despenhadeiro, deixa bater e começa tudo outra vez" (NAUR & RANDELL, 1968).

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.

Comment [C486]: A versao do SWEBOK que voce está descrevendo é a mais nova?

Comment [C487]: Esta frase ficou incomplete. Precisa concluir o pensamento. É preciso falar aqui que nesta conferencia foi criado o termo Eng. SW motivado pela Crise de SW.

Comment [C488]: Coloque uma referencia aqui

Tais necessidades aumentaram a importância e responsabilidades dos especialistas ligados a uma das áreas da computação, conhecida como engenharia de softwareEngenharia de Software. Com isso, a *Institute of Electrical and Electronics Engineers* (IEEE) e a *Association for Computing Machinery* (ACM) conduziram estudos como uma forma dede modo a promoverção ativamente da Engenharia de SoftwareEngenharia de Software como uma profissão desde 1993, definindo as fronteiras que delimitam a Engenharia de softwareEngenharia de Software, e foi chamado deatravés do Corpo de Conhecimento emda Engenharia de SoftwareEngenharia de Software - *Software Engineering Body of Knowledge* (SWEBOK). (SWEBOK, 2004).

Neste capítulo iremos realizar apresentar uma descrição sobre a visão da Engenharia de softwareEngenharia de Software detalhada e apoiada por um processo de desenvolvimento realizado por profissionais, sociedade científica e órgãos públicos que culminou no guia tema do capítulo.

Comment [C489]: Que necessidades? É preciso encadear melhor as ideias.

Comment [C490]: Este parágrafo ficou confuse.

- Deixar claros os limites dae Eengenharia de Ssoftware com respeito a outras disciplinas como ciência da computação, gerência de projetos, engenharia da computação, matemática, entre outros;

Formatted: Indent: Left: 0,25", Hanging: 0,14", Bulleted + Level: 1 + Aligned at: 0,25" + Indent at: 0,5"

Tabela 11.1. Demonstrativo das cCategorias do Conhecimento conforme o SWEBOK

Comment [C491]: Por que esta distribuicao na tabela? Por que nao colocar um embaixo do outro?

Formatted Table

Especializado	

- **Certificação CSDA (Certificação de Associação no Desenvolvimento de Software);**

Comment [C492]: Esta é a traducao correta?

A certificação CSDA oferece os princípios fundamentais para o avanço do profissional de Eengenharia de Ssoftware, disponibilizando uma forte alavanca para experiência estudantil e as reais requisições do mercado de trabalho. CSDA é o primeiro passo para se tornar um *Certified Software Development Professional* (CSDP).

Comment [C493]: Nao gostei deste texto

O SWEBOK utiliza uma organização hierárquica, decompondo todaos aos KA´s em um conjunto de temas com rótulos reconhecíveis pela área de interesse do leitor sobre a Eengenharia de Ssoftware. A figura 11.1 apresenta o corpo de conhecimento do guia, como também seus níveis hierárquicos.

Comment [C494]: Nao gostei deste texto

11.2.2.1. Requisitos de Software

Comment [C495]: Nao usar 4 niveis de numeracao. Neste caso veja o padrao que fonte usar sem numeracao

A liberação é utilizada neste contexto para se referir a entrega de um item de

software estão disponíveis para entrega é freqüentemente necessário para recriar versões específicas de pacotes e os materiais corretos para entrega da versão.

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.

crimes de computador. Respostas sociais, éticas e de legislação estão sendo desenvolvidas para procurar tratar adequadamente cada caso (Koscianski, 2006).

Ribeiro, D.A. Escolha de uma das áreas de Engenharia de SoftwareEngenharia de Software do SWEBOK Centro de Informática – Universidade Federal de Pernambuco (UFPE), Recife – PE – Brasil

12

Gestão de Riscos de Projetos de Software

Luis Alberto Libânio Lima

Gestão dos Riscos é um processo que garante que:

- Situações benéficas serão alcançadas ou têm maior probabilidade de serem alcançadas;

Os trabalhos dos processos não são de evitar os riscos, pois em situações mesmos favoráveis os riscos podem ocorrer, o objetivo da gestão de riscos não é eliminar os riscos, mais gerenciar os riscos abordados em todas as atividades, mais aumentar as oportunidades e diminuir os efeitos que podem comprometer o bom andamento do projeto.

Para ficar mais claro e simples para o entendimento de todos, a gestão de riscos utiliza um conjunto de processos utilizado para identificação dos riscos e oportunidades que eles possam trazer para a organização, estimar qual o impacto potencial causados por esses obstáculos que possam vir surgir no decorrer do projeto, fornecer métodos para tratar esses impactos, para reduzir a chance dessas ações ocorrerem até um nível aceitável ou para atingir essas oportunidades.

A gestão de risco em projetos de software aborda processos e métodos que discutem as praticas práticas de:

Comment [L496]: Falta o índice do Capítulo, conforme solicitado no template.

Comment [L497]: É melhor trocar esse "de" por "EM" porque da forma como está, não está legal esse título. O correto é Gestão de Riscos EM alguma coisa...

Comment [L498]: Está faltando aquela parte introdutória do capítulo, que fala sobre o que vai dizer no capítulo e quais tópicos vão ser abordados ao longo do mesmo.

Comment [ajhol499]: E principalmente contextualiza o tema do capítulo perante o Livro, "de onde vem", "pra onde vai", como começou... (em poucas linhas)

Comment [L500]: Se não colocar esse "que" fica sem ligação os tópicos abaixo com essa frase que você colocou aqui

Comment [L501]: Esse tópico deveria ser uma Introdução do assunto, e é bom ter uma definição formal do que vem a ser Gestão de Riscos. ☺

Comment [ajhol502]: Sugestões:
1.Começar contextualizando o que é risco;
2.Depois justificar pq é relevante gerenciá-los;
3.Enfim, conceituar Gestão de Riscos partindo de, pelo menos, 2 fontes/referências.

Comment [L503]: Reformular essa frase, pois não ficou legal essa escrita.

Comment [ajhol504]: Aqui é "." E começa uma nova frase: " ... ocorrer. O objetivo ..."

Comment [L505]: MAS, sem o "i"

Comment [L506]: Esse "mais" tem o mesmo sentido do primeiro? Se sim, é melhor retirá-lo, porque ele não é necessário na frase. Pode omiti-lo que o sentido fica melhor.

Comment [ajhol507]: Sugestão de reescrita: "Os projetos, pela sua natureza, são um ambiente onde os riscos precisam ser identificados, analisados, planejados e monitorados (gerenciados) num ciclo contínuo,pois mesmo em situações extremamente farvoráveis os riscos podem ocorrer. Um dos principais objetivos da gestão de riscos é manter os riscos sob controle, onde a eliminação é apenas uma das estratégias disponíveis para alcançar este fim."

Comment [L508]: Pode omitir essa palavra, o sentido da frase fica o mesmo, e a escrita fica melhor.

Comment [L509]: Coloca um sinônimo aqui, já que a palavra utiliza apareceu antes dessa.

Comment [L510]: Sem o "S" porque concorda com impacto potencial

Comment [L511]: É interessante você reformular esse parágrafo, o entendimento não está legal.

- **Monitoramento e controle desses riscos presentes –** acompanhamentos dos riscos identificados, possibilitarem a identificação de novos riscos, executarem planos de respostas a riscos.

Todos esses processos envolvidos são constantemente atualizados durante todas as fases do projeto. Eles se interagem entre si e também com outras áreas de conhecimento, pode haver a necessidade da participação de uma ou n pessoas no envolvimento desses processos (gerentes, especialistas, programadores), de acordo com a necessidade do projeto

12.1. Riscos do Projeto

O que são **Riscos**? Seria a primeira idéia ideia a vir em nossa mente quando trabalhamos com gestão de riscos. **Riscos** são fatos, acontecimentos que podem causar perdas, danos ou ganhos a uma empresa. Geralmente a palavra risco está ligada diretamente a perdas ou danos em um projeto, dificilmente atribuímos riscos a ganho. Um processo de desenvolvimento bem elaborado e planejado visa reduzir ao máximo a probabilidade de eventos adversos que possam comprometer o sucesso de seu projeto. A chance de êxodo de um projeto sempre carrega com si a possibilidade de um impacto negativo, desse modo cabe a empresa avaliar, se o risco é considerado aceitável, se ele ocorrer pode trazer graves conseqüências consequências à organização.

Quanto mais bem elaborado os processo de desenvolvimento, menores serão as chances de ocorrência de riscos, infelizmente mesmo em situações favoráveis, os riscos, mesmo que remotamente, eles podem ocorrer. Isso é motivado pelo grande número de fatores envolvidos no processo que podem afetar diretamente o produto final, podemos citar, como por exemplo,s alguns fatores que podem influenciar seu sucesso: uma mudança de sugerida pelo cliente, alteração na sua equipe, mudança na tecnologia de desenvolvimento, e alguma outra alteração alteração, mesmo que justificadas, podem alterar toda programação e o sucesso do seu projeto.

"A gerência de riscos utiliza como base o conceito de exposição de risco. para

Comment [L513]: Está solta esta frase aqui depois da vírgula.

Comment [ajhol514]: Aqui valeria a pena uma figura ilustrando a "dinâmica" deste processo, para que o leitor não fique com a impressão errada que é algo estático, vide (PMBOK, 2008).

Comment [L515]: Que processos envolvidos? Não seriam práticas, conforme você introduz lá em cima? ".... **que discutem as práticas de:"**

Comment [L516]: Não é necessário esse "se", pois já tem o "entre si"

Comment [L517]: Essa frase está solta depois da vírgula? Esse pode seria podendo?

Comment [ajhol518]: Acho que aqui vc inverteu a ordem natural das coisas. Sugiro que conceitue riscos antes de falar dos processos ou etapas do Gerenciamento de Riscos.

Formatted: Font: Bold, Portuguese (Brazil)

Comment [L519]: Reformular essa frase. Tem que ter uma concordância com o que você escreveu antes.. por exemplo, se você faz a pergunta: O que SÃO riscos? Você tem que seguir com o verbo nesse tempo. Por exemplo: É a primeira ideia que vem em nossa mente quando trabalhamos com gestão de riscos...

Comment [L520]: É essa palavra mesmo que você quis dizer??

Comment [L521]: Não está fazendo uma ligação coerente com a frase anterior.

Comment [L522]: Não é necessário esse pronome, já que o sujeito "os riscos" já está explícito na frase.

Comment [L523]: Ficaria mais correto se fosse um ponto final e não uma vírgula e começaria com a palavra "Podemos" com letra maiúscula.

Comment [L524]: Pode omitir essa palavra, fica melhor o texto ☺

Comment [L525]: Sucesso de quem? Do produto final? Do leitor? É melhor escrever de outra forma, porque está dúbio dessa forma.

tamanho da perda (impacto i no projeto, artefato, ativo ou qualquer elemento que seja o foco da análise de risco)" (ESPINHA & SOUSA, 1).

Dessa forma, a finalidade de exposição foi estendida, demonstrando a determinação de um impacto e como se concretiza um risco. Cada membro da empresa que participa do projeto dos processos é atribuído um índice de exposição. Este índice, chamado de psr, ele mostra a rigor a ocorrência do risco(p) para o projeto ou para a(s) empresa(s), a importância das pessoas envolvidas e dos processos, pois esses estão ligados diretamente a qualidade do produto final. Diante disso fica mais claro observar a classificação e o impacto que os riscos podem trazer.

Vamos demonstrar a seguir as principais etapas na utilizadas pela gestão de riscos de projetos de software.

12.2. Planejamento do Gerenciamento de Riscos

O plano de gerenciamento de riscos, aborda como o gerenciamento de risco será tratado e executado dentro de um projeto. Daqui por diante ele passa a ser um subitem gerenciamento do seu projeto, tendo então toda a importância dentro do contexto organizacional.

Essa atividade é de suma importancia importância para o seu projeto, pois ele assegura que o nivelnível, o tipo e a visibilidade do gerenciamento de riscos se enquadre com os riscos e a importancia importância do projeto em relação a empresa, com isso teremos mecanismos para fornecer tempo e recurso suficientes para permitir a execução das atividades de gerenciamente de riscos.

Descreveremos logo abaixo alguns requisitos fundamentais que devem abordos por esses planos 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 duvidas dúvidas quanto a à responsabilidade de cada membro envolvido.

- **Tempos** – avalia quando e com que frenquêencia o gerenciamento de

Comment [L527]: Não está no formato padrão acordado em sala e com o SBC. Deve estar dessa forma: [AUTOR ANO]. Por exemplo: [ESPINHA & SOUSA 2009]

Comment [L528]: Não seria: Para cada membro.....é atribuído um índice de exposição.

Comment [L529]: Esse pronome não é necessário aqui.

Comment [L530]: Frase solta no texto...

Comment [ajhol531]: Importante: vc não vai "demonstrar" nada aqui, pq demonstrar é: "Provar por meio de raciocínio concludente; fazer a demonstração de... (AURELIO, 2005)". Principalmente pq vc está trazendo um conhecimento de outras fontes para apresnetar e não exclusivamente seu. Vc aqui vai no máximo "apresentar", "expor", , "descrever", etc.

Comment [L532]: ???

Comment [L533]: Da forma como está escrita essa frase, parece que todas as seções daqui pra baixo estão dentro das principais etapas uitlizadas pela gestão de riscos. É melhor reformular essa frase.

Comment [L534]: Será que a palavra Gestão não ficaria mais de acordo com o Título do Capítulo?

Comment [L535]: O mesmo comentario acima

Comment [L536]: Num mesmo parágrafo a palavra gerenciamento foi repetida 3 vezes.

Comment [L537]: Um subitem gerenciamento??

Comment [L538]: Não seria "ela" ?? A atividade que assegura?

Comment [L539]: Escrita confusa. O que você quis dizer?

Comment [L540]: ?????

Comment [L541]: Lá em cima você falou: "O plano de gerenciamento de riscos" E aqui você diz: "esses planos de gerenciamentos de riscos" E aí? É um ou são vários?

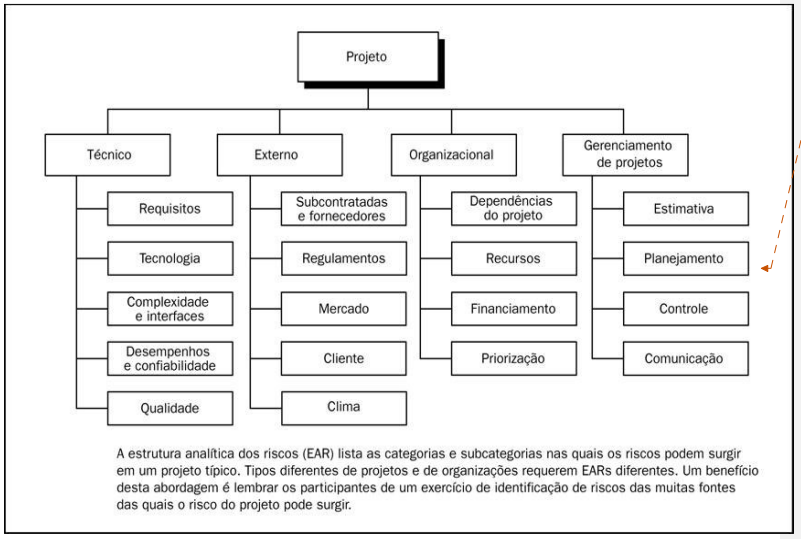
Comment [L542]: Seria melhor no plural, como o verbo "Distribuem" na 2º. Linha abaixo dessa.

Comment [L543]: O verbo "tirar" deveria estar no plural

Comment [L544]: Deve ter o sinal da crase.

Comment [L545]: Não seria melhor no singular?

- **Categorias dos Riscos -** fornecem uma estrutura analitica analítica dos riscos que garante que um processo abrangente para identificar sistematicamente os riscos até um nivel nível consistente de detalhes possibilitando, maios eficácia na qualidade da identificação dos riscos. Uma organização pode usar se uma estrutura similar a essa para categorizar seus riscos, e novas categorias dos riscos podem ser incluidos no decorrer do projeto, visto que esse processo é continuo. Uma boa dica seria revisar as categorias dos riscos antes do processo de planejamento, ou seja, antes de usá-las no processo de identificação dos riscos, logo abaixo podemos observar melhor através da figura a seguir.



A estrutura analítica dos riscos (EAR) lista as categorias e subcategorias nas quais os riscos podem surgir em um projeto típico. Tipos diferentes de projetos e de organizações requerem EARs diferentes. Um benefício desta abordagem é lembrar os participantes de um exercício de identificação de riscos das muitas fontes das quais o risco do projeto pode surgir.

Figura 12-1. Exemplo de uma estrutura analítica dos riscos (EAR).
Fonte: http://wpm.wikidot.com/artefato:plano-de-gerenciamento-de-riscos

- **Definições de Probabilidade e Impacto dos Riscos** – A qualidade e credibilidade do processo analise qualitativa de riscos exigem a definição de niíveéis diferentes de probabilidade e impactos de riscos. As metricas 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 improvavél”

Comment [L546]: Deve estar no plural

Comment [L547]: maior? Essa vírgula antes não existe, pois não se separa verbo do seu complemento.

Comment [L548]: É melhor reescrever este parágrafo.Não está claro.

Comment [L549]: ?? É necessário reescrever esse parágrafo também.. não dá pra entender o que você quis dizer..

Comment [L550]: 1.É com letra maiúscula 2. Seria interessante colocar a numeração da figura aqui.

Formatted: Centered

Formatted: Font: 10 pt, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Comment [L551]: Deve estar centralizado com a Figura. Em tamanho 10 e em negrito.

Formatted: Font: 10 pt, Portuguese (Brazil)

Formatted: Font: 10 pt, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Comment [L552]: É o nome do processo? Se for, seria bom colocar as inciais dele em maiúsculo.

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).

Outra forma de avaliar a probabilidade pode ser feita através de descrições do status do projeto atual relacionado aos riscos que está sendo considerado.

A escala é muito importante ao para o projeto, pois avalia o grau de importância do impacto, como sendo negativa para as ameaças ao projeto e positiva para as oportunidades que possam surgir, em cada escopo do projeto se o fato de risco ocorrer. As escalas de impacto são específicas do objetivo potencialmente afetado, do tipo e do tamanho do projeto, da situação financeira e, das estratégias da organização e da sensibilidade da 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). 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), esses valores não lineares são utilizados quando a empresa deseja evitar ameaças com grande chance de ocorrem, mesmo tendo uma probabilidade baixa de acontece, quando se utiliza essa escala é muito importante você entender o significado de cada número, como eles se interagem entre si com os seus derivados e qual o impacto que ele pode trazer nos objetivos do projeto.

Logo abaixo mostraremos um modelo de como os imprevistos do projeto podem ser usados numa avaliação dos impactos de riscos com relação aos quatros objetivos essenciais em qualquer projeto de software. Esse modelo visa demonstrar os aspectos relacionados a dados numéricos (não-linear). Nesta abordagem não estamos fazendo um paralelo entre números com os termos relativos ao projeto, mais sim exemplificar as duas as alternativas em uma única figura.

- **Matriz de probabilidade e impacto** – as prioridades dos riscos ocorrem e acordam com o grau de impacto de deles nos objetivos do projeto, ou seja, quais as possíveis conseqüências consequências do impacto. “Uma forma simples e de uso freqüente frequente nas organizações, é o uso de uma tabela de pesquisa ou uma matriz com a probabilidade de impactos” (Ferrari, 1). As combinações explicita de probabilidade e o impacto podem fazer com que os impactos adversos possam ser caracterizados como importância: Alta, moderada ou baixa. Ao lado que com essa importância é realizados o **planejamento de respostas aos riscos**, e são desenvolvidas pela empresa. Esse planejamento pode ser revisado durante o projeto e, principalmente, ele pode ser acoplado no **planejamento do gerenciamento de riscos.**

Comment [L553]: É melhor reescrever esse parágrafo.. A escrita não está boa..

Comment [L554]: No parágrafo acima você usa o verbo “Poderia” e aqui você já usa o verbo “Pode”. Seria bom escrever no mesmo tempo verbal.

Comment [L555]: Quem está relacionado a que? Quem está sendo considerado? Confusa essa frase.

Comment [L556]: Não deu pra entender o que você quis dizer... é melhor reescrever esta parte.

Comment [L557]: Referência incompleta e sem padrão

Comment [L558]: O correto é ter um ponto final e começar a palavra Esses com letra maiúscula.

Comment [L559]: ????

Comment [L560]: ???????

Comment [L561]: Deve ser ponto final e começar com letra maiúscula.

Comment [L562]: Entre si com os seus derivados???

Comment [L563]: Ele quem???

Comment [L564]: São os riscos que estão com relação aos quatro objetivos essenciais..???

Comment [L565]: ?? Não entendi..

Comment [L566]: MAS

Comment [L567]: Reescrever esse parágrafo...

Comment [L568]: Esse tópico faz parte de que?

Comment [L569]: Apenas deles sem o “de”

Comment [L570]: É uma pergunta?

Comment [L571]: Referência sem formatação devida.

Comment [L572]: Explícitas?

Comment [L573]: Mudar a escrita dessa parte, porque não está coerente.

Comment [L574]: Reescrever.. parágrafo confuso.

Condições definidas para escalas de impacto de um risco em objetivos importantes do projeto
(os exemplos são mostrados somente para impactos negativos)

	São mostradas escalas relativas ou numéricas				
Objetivo do projeto	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.

Figura 12-2. Definição de escalas de impacto para quatro objetivos do projeto
Fonte: http://wpm.wikidot.com/artefato:plano-de-gerenciamento-de-riscosl

- **Revisão de tolerância das partes interessadas** – podem ser estudadas durante a fase de planejamento do gerenciamento de riscos, pois ela se aplicará a cada projeto mencionado.

- **Acompanhamento** – registra todas as atividades relacionadas as atividades de gerenciamento de riscos, serão guardadas essas informações para uso em beneficio benefício ao do projeto em questão, poderá tirar varias lições, do que foi aprendido e o que erramos. Essa documentação se faz necessaria necessária devido ao um futuro o projeto possa vir passar por uma auditoria e como será feito.

Formatted: Centered

Formatted: Font: (Default) Times New Roman, 10 pt, Portuguese (Brazil)

Formatted: Font: (Default) Times New Roman, 10 pt, Bold, Portuguese (Brazil)

Comment [L575]: Figura? É uma Tabela, não?? Não está referenciada em nenhuma parte do texto.

Formatted: Font: (Default) Times New Roman, 10 pt, Portuguese (Brazil)

Formatted: Font: (Default) Times New Roman, 10 pt, Bold, Portuguese (Brazil)

Formatted: Font: (Default) Times New Roman, 10 pt, Portuguese (Brazil)

Comment [L576]: Quem podem ser estudadas?

Comment [L577]: Deve ter acento grave.

Comment [L578]: Atividades relacionadas às atividades??? Reescreve com outras palavras, pra não ficar desse jeito.

Comment [L579]: ???? Perdido no texto.

Comment [L580]: Reescrever essa parte...falta coerência com o texto.

Comment [L581]: Que documentação?

Figura 12-3 – Reuniões de Planejamento
Fonte: dicas de venda

Participam desse encontro o gerente de projetos, os membros do time de desenvolvimento e as pessoas com interesse no produto final do projeto, alem além de disso, os membros da organização com responsabilidade no gerenciamento de atividades e no planejamento de riscos do projeto.

Durante esses encontros é elaborado o plano de risco para o projeto desenvolvido para executar as atividades de gerenciamento de riscos e definição dos mesmos. Alem Além disso, nessas reuniões são desenvolvidas desenvolvidos os elementos dos custos de riscos e as atividades de programação dos riscos para serem inseridas no orçamento e na programação. São distribuidas distribuídas as responsbilidades 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 resumidas no plano de gerenciamento de riscos". (OSHIRO, 2007).

12.3. Identificação dos Riscos

Peter Drucker [DRU75] disse certa vez: "Embora seja fútil tentar eliminar o risco e

Comment [ajhol583]: Sugiro colocar aqui uma figura que acrescente mais ao conteúdo do capítulo ou da seção. Esta aparentemente serve apenas para ilustração, mas não acrescenta muito. Coloque um diagrama de como conduzir reuniões desta natureza.

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Formatted: Centered

Comment [L584]: Que fonte é essa??? Cadê a referência?

Comment [L585]: Será que pode ter essa Figura aí no meio do nada? Sem ser referenciada no texto...

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Comment [L586]: Tirar o "de"

Comment [L587]: Qual a pontuação certa dessa frase? Quem foi desenvolvido? O plano e risco ou o projeto?

Comment [L588]: O formato correto deveria ser: [OSHIRO 2007] E ela deve constar nas Referências lá na última seção do Capítulo.

Comment [L589]: Falta o ano na referência e ela lá na seção Referências.

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 de sua equipe podem participar dessa atividade quando for solicitada a presença dos envolvidos, desde o gerente de projeto, os membros da equipe, 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.

Comment [L590]: ?? que equipe??

Comment [L591]: Essa parte tem alguma referência ou é de sua autoria mesmo?

Os riscos podem ser classificados em categorias de varias várias formas, iremos adotar uma visão mais ampla do projeto, de tais maneiras que os riscos compreendem***: riscos técnicos, riscos de projeto, riscos de processos e riscos de negócio.***

Comment [L592]: É melhor reescrever esse parágrafo de uma forma mais clara. Não está legal a escrita atual.

- **Riscos Técnicos -** Identificam possíveis falhas que possam vir a ocorrer no projeto, como, por exemplo, na implementação, na manutenção, na natureza da sua codificação, na interface do usuário como o produto e teste. Podem ainda se causado por tecnologia obsoleta, necessitando uma troca na tecnologia a ser desenvolvido o produto. Geralmente, os problemas técnicos podem ocorrer por suas dificuldades em minimizá-los.

Comment [L593]: Como assim???

Comment [L594]: Refaz esse parágrafo.

- **Riscos de Projeto -** Identificam os riscos ligados aos aspectos organizacionais, operacionais e contratuais. Esta atividade é atribuída ao gerente de projeto, este terá o papel de relacionar com fornecedores, com a sua equipe, trabalhar com situações de recursos apertados. Para ficar mais claro, podemos citar um exemplo de uma situação que pode acarretar todo andamento do seu projeto. Atraso no andamento das atividades conseqüentemente consequentemente atraso no seu cronograma causado por um fator externo, por exemplo, falta de comprometimento do fornecedor com a organização, dessa forma se torna complicado o gerenciamento de riscos.

Comment [L595]: É melhor reformular essa parte também. Não está clara.

Comment [L596]: Como assim??

Comment [L597]: Está sem sentido o começo desse parágrafo...

- **Riscos de Negócio** – O risco de negócio é considerado o mais critico crítico dos riscos, pois eles podem destruir todo o seu planejamento e principalmente pode extinguir seu projeto que você considerava o maior de todos os projetos de

Comment [L598]: Eles quem?? Acho que deve ser: Ele pode, já que você começou falando: O risco de negócio.... ou seja, no singular.

Comment [L599]: É melhor reescrever essa parte.

produto fora do escopo da estratégia da empresa, construir um produto no qual a gerencia gerência e a equipe de venda não sabem como vendê-lo.

Quando desejamos agrupar dados para identificar possíveis riscos referentes ao projeto utilizam- se algumas técnicas, ferramentas para auxiliar esse trabalho, descreveremos a abaixo algumas das mais utilizadas para encontrar esses subsídios.

Comment [L601]: As técnicas são as ferramentas para auxiliar esse trabalho?

Comment [L602]: Deve ter um ponto final aqui e começar com letra maiúscula.

Comment [L603]: Que subsídios???

Comment [L604]: Não seria melhor essa palavra no singular não?

12.3.2.1. Revisões da documentação

Pode ser feita uma revisão organizada, estruturada de toda a documentação, planos que envolvem o projeto incluído: planos, premissas, arquivos de projeto anteriores, ou outros dados, qualquer informação pode ser relevante a gestão de riscos. A qualidade dos planos, a consistências das informações, consistências dos planos, pode indicar riscos potencias ao seu projeto.

Comment [L605]: Não está clara essa parte do texto, é melhor reescrever..

Comment [L606]: Como assim?

Algumas técnicas são utilizadas pelas empresas para coleta de dados para serem utilizada pelos especialistas na identificação de possíveis riscos para o seu projeto, logo abaixo descreveremos algumas dessas técnicas:

Comment [L607]: Serem quem? As empresas ou a coleta? Eu acho que é a coleta né? Então é no singular esse verbo.

Comment [L608]: Ponto final e começa com letra maiúcsula.

- ***Brainstorming*** **–** O objetivo dessa técnica é a aquisição de uma lista riscos do projeto. Normalmente ela é empregada por um conjunto de especialistas que não estão ligados ao projeto. Porem Porém utiliza utiliza-se um mediador para demonstrar idéias ideias sobre os riscos do projeto. Pode utilizar a classificação dos riscos como referenciareferência.

Formatted: Font: Italic

- **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 influencias o resultado final.

Comment [L609]: Não está clara essa parte do texto.

- **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.

•**Analise dos Pontos Fortes e Fracos, Oportunidades e Ameaças(SWOT)** - Através dessa técnica certifica-se que o análise realizada no projeto da cada uma das expectativas da análise SWOT, isso faz com que aumente amplitude dos riscos.

Comment [L610]: Tem referência bibliográfica essa parte??

A lista de verificação de identificação de riscos pode ser elaborada a partir das informações de informações passadas ou , com do conhecimento acumulado em outros projetos similares ao atual ou de outras fontes que de informações.

Comment [L611]: ????

O nível inferior da **estrutura analítica dos riscos (EAR)** pode ser utilizado como uma lista de verificação de riscos. Geralmente essa lista e simples e rápida, se torna inviável construir uma lista onde aborda todos os riscos.

Comment [L612]: não está clara essa parte.

"Qualquer projeto de software é desenvolvido a partir de um conjunto de: hipóteses, idéias, cenários ou premissas" (FERRARI). 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.

Comment [L613]: Fora do padrão sugerido, além de não se encontrar na seção de Referências.

As técnicas com diagramas podem ser utilizadas no gerenciamento de riscos da seguinte forma:

Comment [L614]: Num é no plural, conforme o título da seção??

- **Diagrama de Causa Efeito** – ela é realizada para controle de qualidade, esse diagrama, também chamado de diagramas de Ishikawa ou diagramas espinha de peixe, demonstra como os diversos fatores podem esta estar ligados a possíveis problemas.

Comment [L615]: É melhor tirar essa parte e colocar apenas o verbo: Realizado para controle de qualidade, esse diagrama....

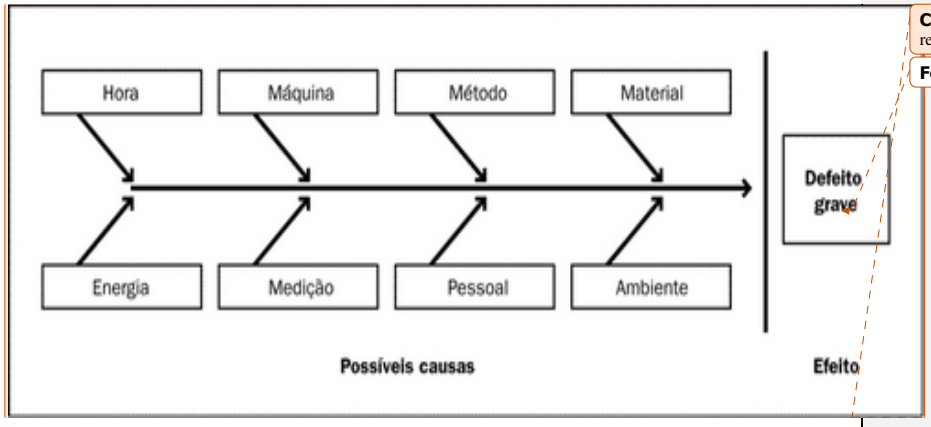


Figura 12-4- Possíveis causas efeito

Fonte: http://wpm.wikidot.com/tecnica:diagrama-de-causa-e-efeito

- **Diagrama do Sistema ou Fluxograma –** auxilia no processo de detalhamento do problema. Os fluxogramas são utilizados para detalhamento do processo. Existem diversos fluxogramas, mais em sua maioria demonstram as atividades, os pontos de decisão e a ordem de processamento. Alem Além disso eles detalham o inter-relacionamento com todos os processos envolvidos no projeto. Veja a baixo um fluxograma utilizado na revisão de processos. Podem ser notados nos fluxogramas alguns problemas de qualidade que podem ocorrer, e aonde eles podem surgir.

Comment [L616]: A Figura deve ser referenciada no texto em algum momento.

Formatted: Centered

Comment [L617]: Esse nome dessa figura deveria ser Diagrama Causa-Efeito.

Formatted: Font: 11 pt, Bold, Portuguese (Brazil)

Formatted: Font: Bold, Portuguese (Brazil)

Formatted: Font: 11 pt, Bold, Portuguese (Brazil)

Formatted: Font: 11 pt, Bold, No underline, Font color: Auto

Formatted: Font: Bold, Portuguese (Brazil)

Comment [L618]: Você diz uma coisa, depois você diz outra... E aí? Detalhamento do problema ou do processo?

Comment [L619]: Onde mostra isso na figura?

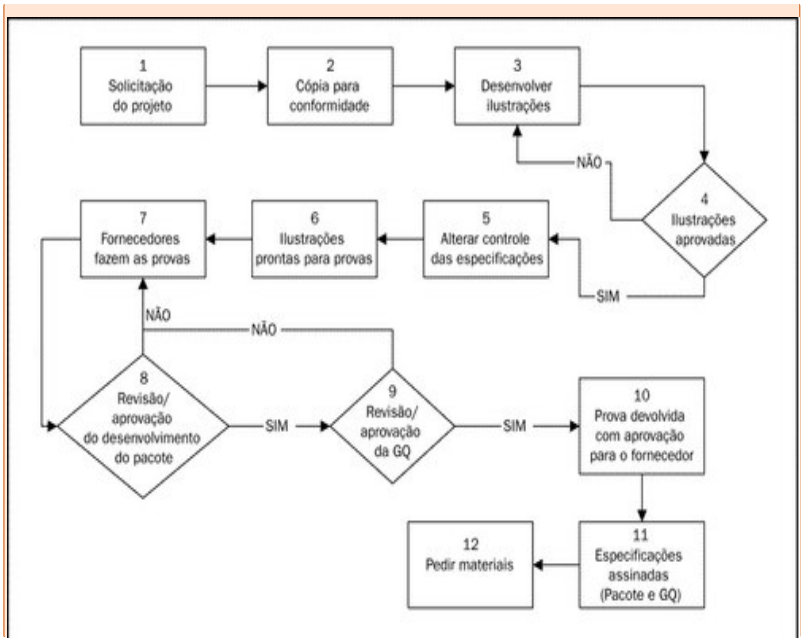


Figura 12-5 - Fluxograma
Fonte: http://wpm.wikidot.com/tecnica:elaboracao-de-fluxogramas

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. Esta fase realiza avaliação dos riscos identificados de acordo com sua probabilidade deles ocorrerem, com o impacto causado no escopo do projeto se eles vierem ocorrer, alem além de outros fatores como: o prazo do risco, a tolerância riscos as restrições de custos, cronograma e na qualidade do projeto.

Os significados dos níveis das perspectivas e impacto, as entrevistas com especialistas auxiliam no processo de correção de desvios sistemáticos comumente apresentados nas informações utilizadas nesse processo. As atitudes criteriosas no prazo das tarefas relacionadas aos riscos podem aumentar sua relevância. Uma analise análise realizada nos dados com relação aos riscos do projeto também se torna importante para compreender a avaliação da importância de dos riscos para o projeto.

Comment [L620]: Deve ser referenciada no texto, com a sua identificação.

Formatted: Centered

Formatted: Font: 10 pt

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Comment [L621]: Qual a pontuação desse parágrafo? Está confuso.

Comment [L622]: Quem vierem a ocorrer?

Comment [L623]: Não está bem escrita essa parte. É melhor reescrever..

Comment [L624]: Não está clara essa parte.

Comment [L625]: Não entendi..

Comment [L626]: Não tem esse “de”

aos riscos, alem além de instituir métricas para analise análise quantitativa dos riscos, essa etapa só e utilizada quando solicitada. Durante todo o projeto analise análise qualitativa dos riscos deve ser reavaliada, para acompanhar as variações dos riscos nos riscos do projeto.

Comment [L627]: Que etapa?

Comment [L628]: É uma fase ou um projeto??

Comment [L629]: Dos riscos nos riscos.... não tá legal essa parte.

Nesta fase da analise análise qualitativa dos riscos, se empregam-se métodos para avaliação de probabilidade caos caso os riscos venham a ocorrer. Avaliação do impacto do risco ocasionado elabora os efeitos sentidos no escopo do projeto, no objetivo, aspectos financeiros (custos), qualidades, alem das ameaças (efeitos negativos) e também os riscos que possam trazer oportunidades (positivos).

Comment [L630]: Reescrever essa parte, pois não está com a escrita clara.

A probabilidade e a força dos riscos são avaliados individualmente para cada risco identificado. Eles podem ser avaliados de diversas formas desde a partir de simples reuniões com sua equipe, com entrevistas com membros do projeto, até as sugestões de especialistas não ligados ao projeto. A avaliação de 3º, ou seja, de especialistas não desligados do projeto se faz necessário, pois, em muitos casos a 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.

Comment [L631]: ???????

"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). 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, mais mas mesmo nesses casos os riscos devem ser monitorados e controlados.

Comment [L632]: Referência sem o padrão devido e sem estar presente na seção Referências

12.4.2 Matriz de Probabilidade e Impacto

"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). Essa matriz relaciona o conjunto das combinações de probabilidade e com o impacto que venha qualificar os riscos como: prioridade baixa, moderada e alta.

Comment [L633]: Referência já comentada acima.

Comment [L634]: É o impacto que qualifica os riscos??

As empresas devem associar as combinações de probabilidade e impacto que implica em: risco alto ("condição vermelha"), risco moderado ("condição amarela") e

Comment [L635]: Deve ter a numeração dela também aqui.

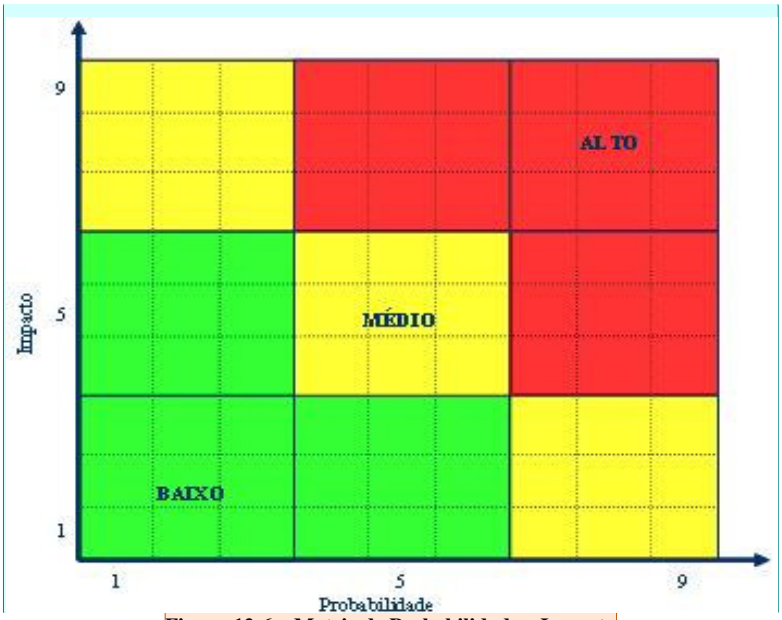


Figura 12-6 – Matriz de Probabilidade e Impacto
Fonte: http://www.tenstep.com.br/br/imagens/Bezerra3.JPG

Por convenção, geralmente são as organizações que estabelecem as premissas de riscos antes do projeto. E são anexadas no ativo dos processos organizacionais. Essas regras podem ser ajustadas no processo de planejamento do gerenciamento dos riscos para o projeto atual.

Uma organização pode classificar seus riscos de acordo com seus objetivos organizacionais: **custo, prazo e escopo.** Alem Além disso, pode estabelecer uma medida única para qualificar de forma ampla os riscos. Por último, você pode avaliar as oportunidades e ameaças nesta mesma matriz utilizando fatores de vários níveis de impacto que são ajustadas para cada uma. Os pontos dos riscos auxiliam a orientar as respostas dos riscos. Imagine a seguinte situação: os riscos, que se vierem a ocorrer, podem trazer efeitos negativos ao seu projeto caracterizando as ameaças ao seu projeto e se localizam em regiões de alto risco, isso pode levar a organização a montar estratégias, mecanismos de urgência para combater esses possíveis riscos. As ameaças caracterizadas como de risco baixo, não devem exigir muito trabalho, o ideal e é você selecionar essas esses riscos e colocar em uma lista apenas para monitoramento e controle para seu acompanhamento. Semelhantes aos efeitos negativos (ameaças), as oportunidades também devem ser priorizadas, primeiro você deve selecionar as oportunidades consideradas de alto risco, pois essa agregará mais aos objetivos da organização, as oportunidades de baixo risco devem ser somente monitoradas.

Comment [ajhol637]: Qualidade da Figura não está boa para impressão. Texto dos eixos fica praticamente ilegível.

Comment [L638]: Tem que utilizar tons de cinza na figura.

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Formatted: Centered

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Comment [L639]: Reescrever essa parte, pois como você colocou ponto, ficou sem nexo. E mesmo que colocar vírgula, deve ser escrito de outra forma, porque esta atual não cabe aí.

Comment [L640]: Que regras??

Comment [L641]: O que são ajustadas para cada uma? Cada uma quem?

Comment [L642]: Reescrever de outra forma. Não está correta dessa forma.

Comment [L643]: Repetiu na mesma frase ao seu projeto duas vezes. Trocar o termo por outro.

Comment [L644]: Reescrever esse texto também, não está coerente com a frase anterior..

Comment [L645]: Trocar esse termo, não está correto com a frase.

Comment [L646]: Ponto final e começa com maiúscula.

Comment [L647]: Ponto final, começa com maiúscula.

Comment [L648]: Mais o que?

Uma analise 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 a de integridade desses eventos. Informação com pouca qualidade implica em numa analise 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 aos dos envolvidos, pois os dados podem afetar diretamente ao seu projeto final.

Comment [L650]: Ponto final e começa com maiúscula.

Os riscos em um projeto de software podem ser relacionados de acordo com a origem dos riscos, usando como base o modelo proposto do EAR, ou simplesmente pelas áreas comprometidas pelos riscos, como é descrito no modelo EAP ou por outra hierarquia útil, como podemos citar uma ciclo do projeto, para expor quais as regiões mais expostas às ameaças dos riscos. A classificação e a junção dos riscos de mesmas grandezas podem facilitar no desenvolvimento de respostas aos riscos mais eficientes e eficazes.

Comment [L651]: Reescrever essa parte. Não está clara.

Dentre os vários riscos identificados, sempre haverá alguns tratados com maior prioridade do que os demais, então havendo necessidade a abordagem dos riscos que exigem premissas que possam vir a evitar, solucionar exigindo uma resposta imediatamente esse com certeza terão que ser tratado o mais rápido possível.

Comment [L652]: Não está compreensível este parágrafo. É necessário reescrevê-lo

Alguns fatores podem indicar sua prioridade na avaliação como podemos citar: o tempo necessário para apresentar uma solução aos riscos, os sintomas dos riscos, a emissão de sinais de alertas e principalmente sua classificação.

Comment [L653]: Prioridade de quem? Não está no texto isso.. esclarecer com outras palavras.

Comment [L654]: ??

Este processe processo tem por finalidade avaliar numericamente a possibilidade de cada ameaça ocorrer e suas possíveis conseqüências consequências junto ao objetivo do projeto, alem além do objetivo geral do projeto. Esta etapa usa-se de algumas técnicas

Comment [L655]: É um processo ou é uma fase, como dita acima no começo??

Comment [L656]: Como assim?

Assim com na analise análise qualitativa dos riscos a abordagem de analise análise quantitativa dos riscos requer antes de tudo a identificação dos riscos. As duas analise análises podem ser realizadas paralelamente uma da outra ou podem ser feitas diretamente juntas. "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).

Comment [L657]: Não são necessárias essas palavras

Comment [L658]: Como assim? Diretamente juntas?

Comment [L659]: Referência deveria estar dessa forma: [PMBOK 2004]

- **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 evidencia evidência são baixos, ou numa situação pessimista, ou seja, aqueles riscos são fortes candidatos a ocorrem r e prejudicar seu projeto. Algumas metodologias utilizam uma abordagem diferente no que se referem, alguns utilizam a media média dos riscos identificados por outro lado outros calculam de acordo com o desvio padrão. Todas as informações do projeto, como a documentação da analise análise lógica da fase de riscos, as informações adicionais sobres eventos adversos, são subsídios de extrema relevância para as entrevistas sobre gerenciamento de riscos, neles pode haver dados confiáveis e de credibilidade para analiseanálise.

  Vejamos a baixo uma figura que mostra um exemplo de estimativas em 3 pontos para a estimativa de custos:

Comment [L660]: Referência já comentada acima.

Comment [L661]: No que se referem a que?

Comment [L662]: Mal escrita essa parte.

Comment [L663]: Neles quem??

Comment [L664]: A Figura deve estar identificada aqui no texto e com a primeira letra maiúscula.

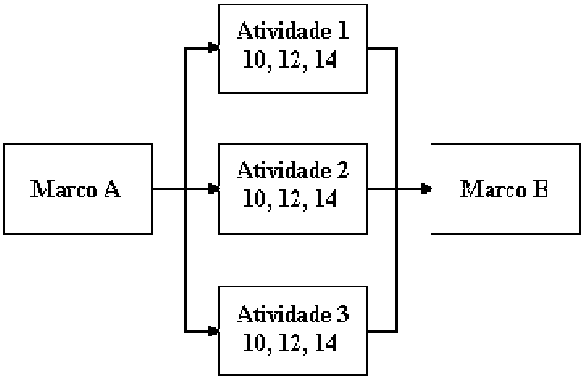


Figura: 12-7, Estimativa em 3 pontos para estimativa de custos.
Fonte: PMBOK

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto

Formatted: Font: 10 pt, Bold

Comment [L665]: Cadê a referência?

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto

Formatted: Centered

Formatted: Font: 10 pt, Bold

A figura a baixo demonstrará duas das distribuições, onde o eixo vertical representa a expectativa e o eixo horizontal o impacto.

Comment [L666]: A Figura deve estar identificada aqui no texto e com a primeira letra maiúscula.

Comment [ajhol667]: Já mencionei o cuidado com o uso da palavra "demonstrar" mais acima...

Comment [L668]: No futuro não, é no presente que deve ser utilizado o verbo.

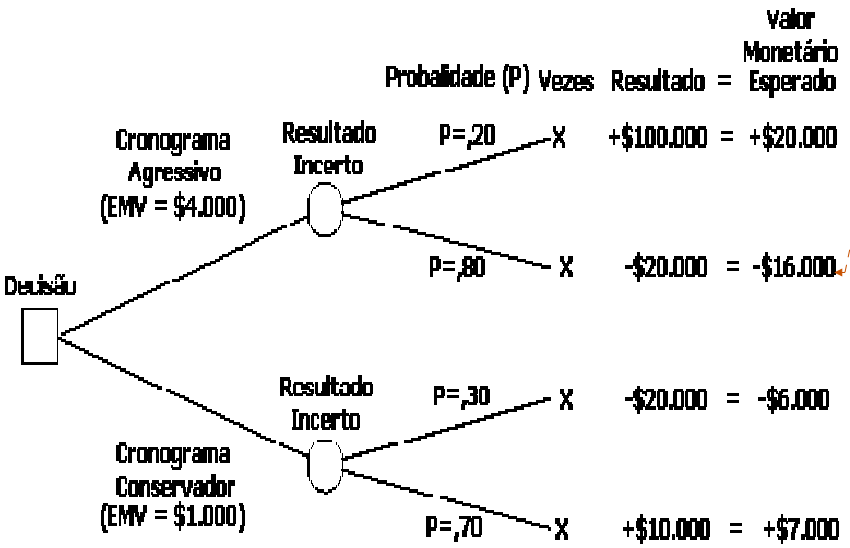


Figura: 12-8, Árvore de decisão
Fonte: PMBOK

- **Analise Análise Sensitiva** – essa etapa auxilia o gerenciamento de riscos na identificação dos riscos com maior impacto no objetivo do projeto. Ele verifica a expansão a incerteza de cada item do projeto afeta aos objetivos que está sendo avaliados quanto todos os requisitos incertos são mantidos em seus valores iniciais.
- **Analise 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" [(PMBOK, 2004)]. Ela anexa as perspectivas dos riscos e os custos de recompensas da de cada abertura lógica dos eventos e de decisões futuras. Solucionando a árvores de decisão indicará quais decisões geram as estimativas maiores para o tomador de decisão

Formatted: Centered

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Comment [L669]: Cadê a referência?

Formatted: Font: 10 pt, Bold, No underline, Font color: Auto, Portuguese (Brazil)

Formatted: Font: 10 pt, Bold, Portuguese (Brazil)

Comment [L670]: Ele quem?

Comment [L671]: Não está claro o entendimento dessa parte.

- **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 de uma função de probabilidade selecionada durante cada interação.

Comment [L672]: Referência comentada acima.

- Etapa de Abrangência – representa um modelo mais compreensível e menos rigorosa rigoroso da implementação dos modelos e das normas de referenciareferência, com isso identificar quais as áreas da organização são mais vulneráveis a riscos. O objetivo seria de localizar o problema.

Comment [L673]: Está sem nexo essa frase aqui.

- Etapa de Profundidade – nessa etapa são decididos quais os campos devem ser o foco de uma avaliação mais criteriosa e da próxima interação do ciclo de melhoria qual é o problema e de qual forma ele pode ser solucionado.

Comment [L674]: Qual a pontuação dessa frase?

Em cada uma dessas fases, são desempenhadas etapas de diagnóstico e priorização e definição dos planos de ação. Na fase de abrangência o diagnóstico é mais amplo e os planos de ação venham se tornar menos detalhados, na etapa de profundidade à análise é mais especifica e os planos se tornam mais detalhados.

O objetivo desta estratégia é complementar métodos como scampiSCAMPI, maturity mps e iso/iecISO/IEC 15504, oferecendo propostas de soluções a alguns potenciais problemas encontrados na aplicação destes métodos (KHAMIANOVA, KREJCI, & OLIVEIRA). Descreveremos abaixo os princípios que rege essa estratégia:

O último principio princípio tem por finalidade demonstrar informações de um nível de abstração granular para o processo de tomada de decisão. Embora a utilização da capacidade de processo e do nível de maturidade seja o parâmetro mais utilizado no direcionamento de recursos na área de desenvolvimento de software, estes conceitos são um tanto abstratos e em muitos casos dificultam esta atividade (se vários processos apresentam a mesma capacidade e o mesmo *gap* entre a capacidade esperada e a

Comment [L675]: Mudar essa parte

Comment [L676]: É isso mesmo?

Comment [L677]: ???

Comment [L678]: Referencia sem padrão correto. E não se encontra na seção Referências.

Comment [L679]: Verbo deve ser no presente e não no futuro.

A estimativa dos riscos tenta classificar os riscos de duas formas: **probabilidade** dos riscos ocorrer e as **conseqüências** **consequências** que ele pode trazer caso ele venha acontecer. São executadas 4 quatro atividades de projeção das atividades, essas tarefas são executadas pelo gerente de projeto:

Comment [L681]: Atividades de projeção das atividades??? É melhor alterar essa frase.

Comment [L682]: É melhor reescrever essa parte e colocasr essa informação na frase anterior.

- As **conseqüências** **consequências** do risco.

Comment [L683]: Isso é uma atividade?

- **Anotação** da precisão global da projeção dos riscos, para que não haja efeitos contrários.

Comment [L684]: É bom seguir algum padrão nesses tópicos. Se o primeiro você fez com travessão, os outros devem seguir a mesma linha.

Uma escala pode definida em termos booleanos, qualitativos e quantitativos (PRESSMAN, 3)

Comment [L685]: Referência já comentada

Elaborando um questionário, com algumas das perguntas abaixo:

Comment [L686]: ??? Solto???

Exemplo: o valor **0,90**, quando atribuído a um risco, determina que ele é considerado altamente provável que ele ocorra.

Comment [L687]: Isso se você deixar claro que a faixa é até 0 a 1...

Os riscos são alocados em função do possível impacto que ele possa trazer ao projeto, logo em seguida são inseridos numa ordem de prioridade. Três fatores e fundamental para afetar o impacto do risco no projeto.

Comment [L688]: ???? e fundamental ???

14.1.5. Exercícios

Comment [L689]: Tem que ter no mínimo 10 questões.

14.1.6. Fontes de Pesquisas

Comment [ajhol690]: Onde estão as “sugestões de leitura”, ITEM OBRIGATÓRIO do capítulo?

Comment [L691]: Você quis dizer: Tópicos de Pesquisa? É obrigatório ter essa seção no Capítulo.

Comment [L692]: Links soltos
Não está no padrão combinado e como no template que o professor recomendou.

Qualidade de Software – Engenharia de Software Magazine.

Comment [ajhol693]: Além das observações de Thaysa, considero: muito poucas referências para um capítulo de um livro!

Comment [L694]:
1.Cadê as referências que estão ao longo do texto:
(ESPINHA & SOUSA, 1).
(FERRARI)
(Ferrari, 1)
(OSHIRO, 2007)
(PMBOK, 2004).
(FERRARI, 2004).
(KHAMIANOVA, KREJCI, & OLIVEIRA).
(PRESSMAN, 3)
[DRU75]

2. Sem formatação alguma
3.Estão inconsistentes com o descrito na seção **14.1.7. Referencias Bibliográficas.**

5.1.1. Como Liderar Reuniões

Comment [G695]: O item “Como liderar reuniões” poderia ser comentado no item anterior “O papel do líder”.

7.6. Como Gerir Conflitos no Ambiente do Projeto

Comment [G696]: O item 7.6 pode ser abordado no item dentro do item 7.3.

1.4. Desafios Organizacionais

Comment [G697]: Verificar a numeração.

2.2. O que significa motivação

Comment [G699]: Mudança do subtítulo: Definição de motivação

Tanto motivação como quanto emoção vêm 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, afirma que uma pessoa seja capaz de motivar outra. Motivação é conseqüência de necessidades não satisfeitas. Essas necessidades são intrínsecas as pessoas. Isso significa que os gerentes não são capazes de motivar, mas de satisfazer às necessidades humanas ou contrafazê-las. (Archer, 1990, p.8)

Comment [G700]: Verificar as referências de acordo com otemplate

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.

Comment [G701]: Fazer uma ligação desta seção com a próxima.

2.2.2. Teoria Humanística da Hierarquia das Necessidades de Maslow

Comment [G702]: Criar um item para as teorias e não detalhá-las e sim expor de uma maneira geral.

atividade social, como amizades, aceitação social, suporte familiar e amor.

2.3 Processos de Motivação

Comment [G703]: Detalhá-las e sim expor de uma maneira geral.

A formação de uma equipe de projetos não acontece somente pela união das pessoas visando à realização de uma dada tarefa. Assim, para se construir uma relação firme é preciso a 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.

Comment [G704]: Fazer uma ligação com a

Nas equipes de projetos tradicionais do aprendizado é pouco recompensado [não ficou claro esse trecho], pois as pessoas têm 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 [G705]: Esse término não tem relação com que estava escrito anteriormente.

Enquanto que na equipe de projeto de auto desempenho acontece o contrário, assim como na estrutura organizacional, onde existem apenas alguns níveis de gerenciamento entre gerencia 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 mesmo. 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.

Comment [G706]: Explicar o que vem a ser uma organização horizontalizada.

Equipes são alicerces das organizações de alto desempenho. Por mais que tentemos, é impossível chegar ao alto desempenho sem as mesmas. A escolha do tipo certo de equipe não é 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 auto-gerenciadas, 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.

Comment [G707]: Padronizar a referência e apresenta poucas citações.

- **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 na maioria das organizações, fazendo pesquisas, fabricando produtos, vendendo, prestando serviços 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.

Comment [G708]: Evitar o uso de muitos marcadores e se utilizar padronizar os marcadores.

- **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

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.

Comment [G709]: Fazer a ligação com a próxima seção.

5.1.1. Como Liderar Reuniões

Comment [G710]: O item "Como liderar reuniões" poderia ser comentado no item anterior

7.6. Como Gerir Conflitos no Ambiente do Projeto

Comment [G711]: O item 7.6 pode ser abordado no item dentro do item 7.3.

7.6.6. Negociar

Comment [G712]: Poderia explicar comentar de uma forma geral, sem dividi-los em itens.

8. Gestão de Pessoas e Desenvolvimento da Inteligência Emocional

GIL, A.C. Gestão de Pessoas. São Paulo: Atlas,2009.

Índice do Capítulo

Comment [h713]: Durante a escrita do capítulo manteremos um índice para o emsmo.

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. Neste capítulo abordaremos inicialmente questões ligadas 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.

Comment [G714]:

Comment [G715]: Como ‘e logo na introducao achei q isso ficou meio repetitivo / quem sabe tirar esta parte e por: Inicialmente será discorrido.../ , ou algum sinonimo

Comment [G716]: Acho que aki precisaria de uma virgula

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 conseqüentemente da organização, porque “contamina” as pessoas com a alegria e o otimismo.

Comment [G717]: Naum sei se seria a melhor palavra ser usada, pois a comunicacao naum faz somente isso, ela vai alem, logo, talvez seria melhor

Os projetos das organizações são realizados por pessoas, as quais necessitam

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.

Tabela 1. Funções da comunicação na organização.

Comment [G718]: Seguir padrao de formatacao

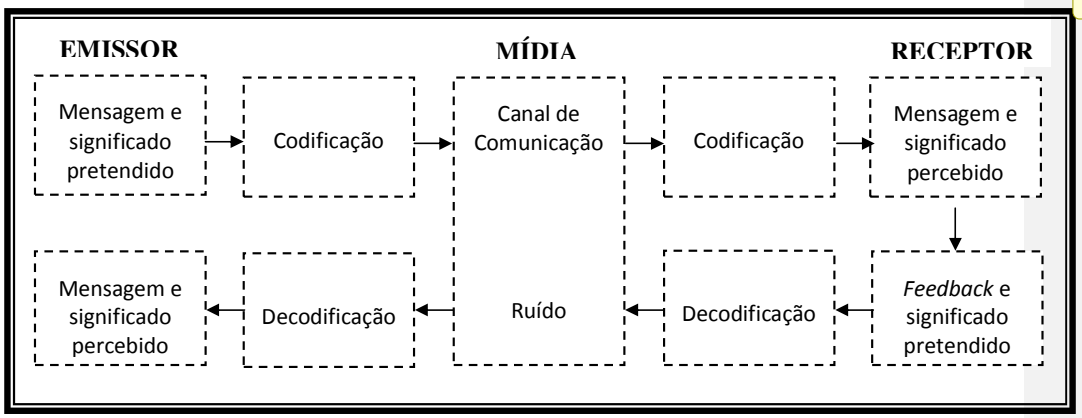
	necessidades sociais.

Fonte: Elaboração própria.

Comment [G719]: Seguir padrao de formatacao

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 1 [Cavalieri 2005].

Comment [G720]: Espacamento entre e texto e figura



Comment [G721]: Tirar espaco e seguir padrao de formatacao

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.

Comment [G722]: Acho que aqui faltaria um O.

Comment [G723]: Esta frase ficou meio confusa, talvez uma virgule aki, e usar algo como , para que

Comment [G724]: Se usar o para aki teria que mudar para: resulte

Receptor da mensagem representa quem recebe e decodifica a mensagem. Aqui é necessário ter atenção que a decodificação da mensagem resulta naquilo que efetivamente o 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

que ocorrerem 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. Sem *feedback*, o emissor não sabe se sua mensagem foi recebida e compreendida.

Comment [G725]: Como vc cita isso na frase anterior, naum sei se seria necessario resptir aki

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.

Comment [G726]: italico

Formatted: Font: Italic

Formatted: Font: Italic

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

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.

De acordo com Verzuh [Verzuh 2000], para a divulgação de todas as

projeto. Algumas reuniões são chaves para a integração da equipe e acompanhamento do projeto. Dentre algumas reuniões merece destaque:

Gerenciar comunicação em projetos é um processo tão importante quanto qualquer outro processo nas empresas. Reconhecer a comunicação como um processo, conhecendo seus elementos, as formas de comunicação e partes envolvidas, é o primeiro passo para implantação de um sistema de gestão eficiente. Um projeto pode

mercado, quando este se torna um ativo que pode ser reutilizado pela empresa na gestão de outros projetos.

As comunicações do projeto sempre foram e continuarão sendo um ingrediente importante na fórmula para o seu sucesso. O PMBoK[13] 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 2.1.1. Apesar de um projeto ser 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 fFigura 2.

Comment [G727]: Organizar figura e texto Escondido atrás.

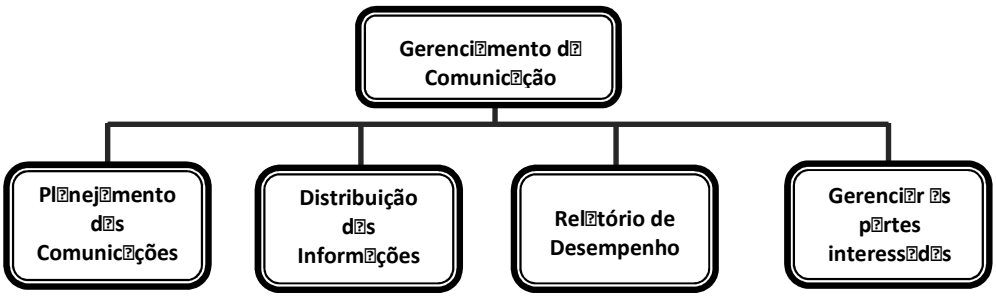


Figura 2. Visão geral do gerenciamento das comunicações do projeto.

Comment [G728]: formatar

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.

Os processos conforme mostrados na Ffigura 2 se relacionam e interagem durante a condução do projeto, a descrição de cada um desses processos é feita por meio

Comment [G729]: Seguir padrao de formatacao

[13] PMBOK é um *Guide to the Project Management Body of Knowledge* desenvolvido pelo *Project Management*

dos seguintes termos: *Entrada:* são representadas por documentos, planos, desenhos; *Ferramentas e Técnicas*: são aplicadas as entradas; S*aídas*: são representadas por documentos, resultados, produtos.

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 continua. Esse planejamento é freqüente e firmemente relacionado ao planejamento organizacional, visto que a estrutura organizacional do projeto terá um maior efeito nos requerimentos de comunicação. A composição do processo do planejamento das comunicações pode ser visualizada na Ffigura 3.

Comment [G730]: Organizar a figura

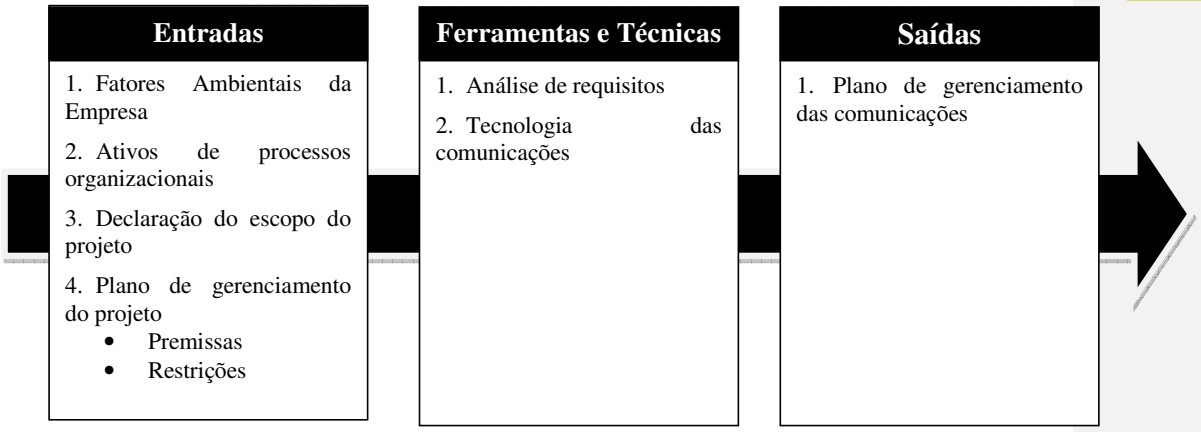


Figura 3. Planejamento das comunicações.

Comment [G731]: Seguir padrao de formatacao

14.2.1.1. Entradas para o Planejamento das comunicações:

Ao longo do desenvolvimento do planejamento da comunicação e da documentação subseqüente do projeto, todos e quaisquer ativos 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.

Os ativos de processos organizacionais 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. Os ativos de processos organizacionais podem ser organizados de diversas formas, dependendo do tipo de setor, organização e área de aplicação [PMBOK 2004]. Por exemplo, os ativos de processos organizacionais poderiam ser agrupados em duas categorias: Processos e procedimentos da organização para realizar o trabalho eBase de conhecimento corporativo da empresa para armazenar e recuperar informações.

Comment [G732]:

Comment [G733]: Sugiro mudar em um dos lugares, pois fica repetitivo , a menos que seja este o objetivo, para dar enfase

Comment [G734]:

Comment [G735]: Para nau mrepetir, talvez aki poderia usar, estes

Comment [G736]: Repetido novamente

A declaração do escopo do projeto descreve detalhadamente as entregas do projeto e o trabalho necessário para criar as entregas. A declaração do escopo do projeto 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.

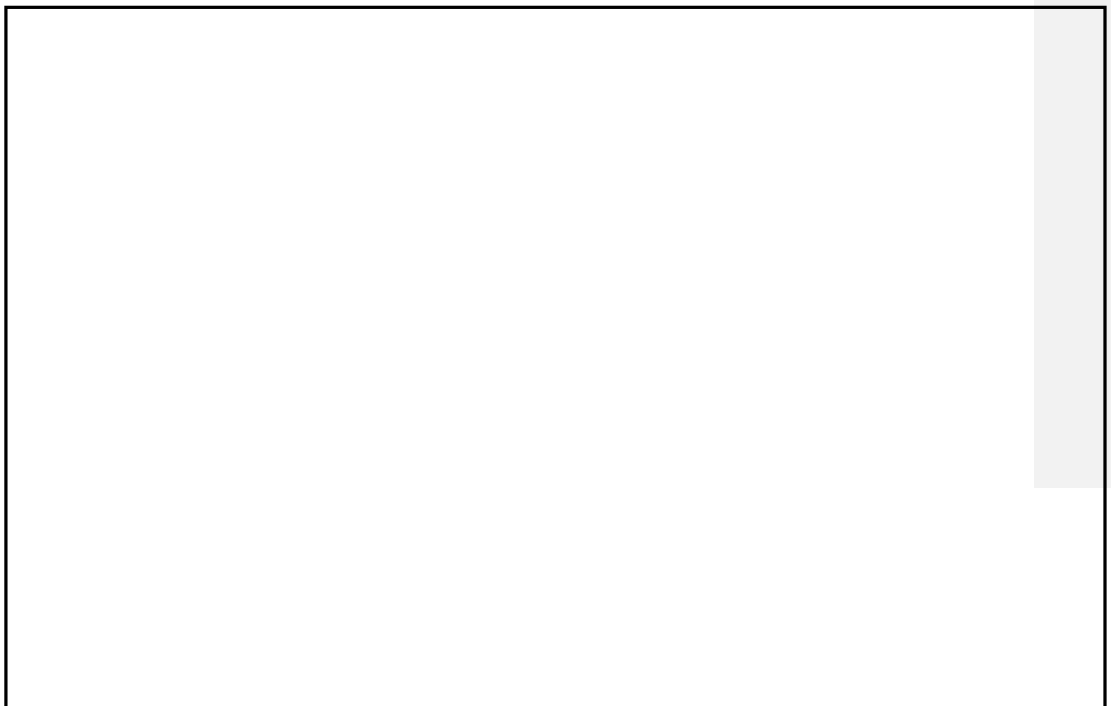
O nível e o grau de detalhamento com que a declaração do escopo do projeto define o trabalho que será realizado, bem como o que será excluído podem determinar a eficácia com que a equipe de gerenciamento de projetos poderá controlar o escopo global do projeto. O gerenciamento do escopo do projeto, por sua vez, pode determinar

Comment [G737]:

Comment [G738]:

declaração do escopo do projeto inclui, diretamente ou referencia outros documentos como:

O 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 planejamento[14] e inclui:



Template 1. Modelo do plano de gerenciamento de comunicações.

Comment [G740]: Seguir padrao de formatcao

Figura 4. Tipos de informações que devem ser distribuídas.

Comment [G741]: Seguir padrao de fornatacao

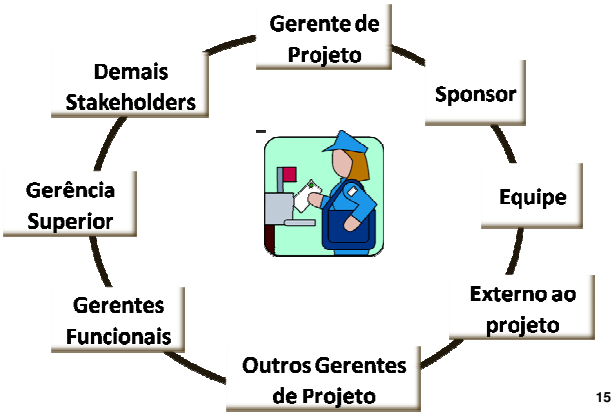


Figura 5. Para quem as informações que devem ser distribuídas.

Comment [G742]: Mesmo que os anteriores

Entradas	Ferramentas e Técnicas	Saídas
1. Plano de gerenciamento das comunicações	1. Habilidades de comunicação 2. Sistema de coleta e recuperação das informações	1. Ativos processos organizacionais (atualizações) 2. Mudanças solicitadas

Figura 6. Distribuição das informações.

Comment [G743]: Comentado nas outras figuras

Conforme descrito na seção 14.2.1.3.

Comment [G744]: Essa parte voce ira colocar mais algo? Se naum será que e preciso deixa-la, achei meio esquisito ela assim apenas sendo citada

1. ***Habilidade de comunicação***

Comment [G745]:

Comment [G747]:

Comment [G748]:

Comment [G746R745]: Trazer para o inicio desta pagina

As habilidades de comunicação fazem parte das habilidades de gerenciamento geral e são usadas para trocar informações. As habilidades de gerenciamento geral estão relacionadas às comunicações e tem como objetivo garantir que as pessoas certas obtenham as informações corretas no seu devido momento, conforme definido no plano de gerenciamento das comunicações. Além disso, essas habilidades também incluem a arte de gerenciar os requisitos das partes interessadas.

O processo de lições aprendidas se concentra na identificação dos sucessos e fracassos do projeto, incluindo sugestões 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 [G749]: Falta uma preposicao para linkar a frase

- *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.

Formatted: Font: Italic

Comment [G750]: Diminuir intervalo

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 mudanças[16].

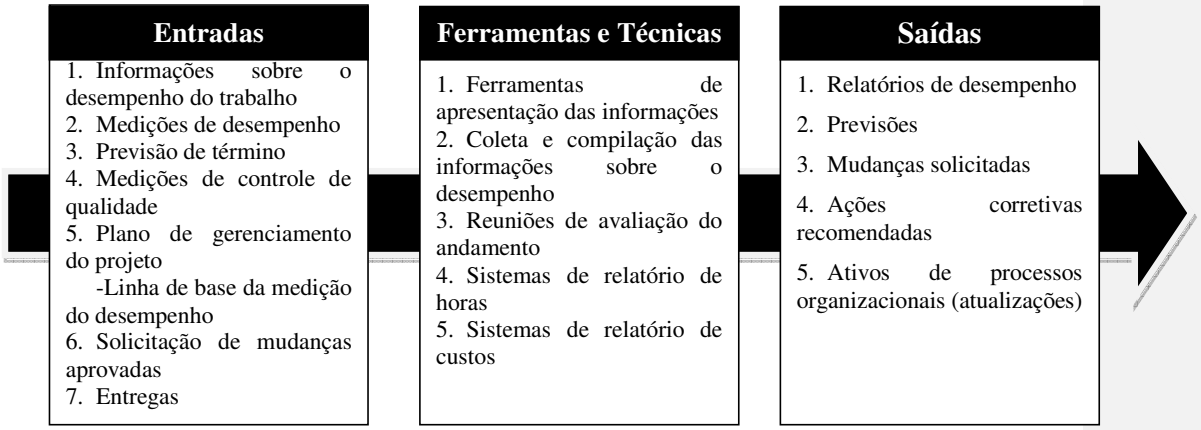


Figura 7. Relatório de desempenho.

Comment [G751]: Formatar

Tabela 2. Valores-chave da técnica do valor agregado.

Comment [G752]: Formatar fonte e espacamento

		Custo orçado do trabalho agendado a ser terminado em uma atividade ou o componente da EAP[17] até um determinado momento.
Índice de		O IDC é o indicador de eficiência de custos mais

desempenho de custos		comumente usado, sendo assim, um valor de IDC < 1.0 indica um estouro nos custos estimados. Um valor de IDC > 1.0 indica custos estimados não atingidos. O IDC é igual à relação entre VA e CR.. Fórmula: IDC = VA/CR

4. Medições de controle de qualidade

Comment [G753]: formatar espacamento

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)[18] 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.

É qualquer produto, resultado ou capacidade para realizar serviços exclusivos e verificáveis que devem ser produzidos para terminar um processo, fase ou projeto. O termo é freqüentemente utilizado mais especificamente com referência a uma entrega externa que está sujeita à aprovação do patrocinador ou cliente do projeto.

Comment [G754]: nova grafia

4. Sistemas de relatórios de horas

Comment [G755]: espaco

Os sistemas de relatórios de horas registram e fornecem as horas gastas no projeto.

Figura 8. Relatório de desempenho gráfico (ilustrativo).

Comment [G756]: formatar

	Planejado	Agregado	Custo					Índice de desempenho	
Elemento da EAP	Orçamento	Valor agregado	Custo real	Variação de custos		Variação de prazos		Custo	Cronograma
	($) (VP)	($) (VA)	($) (CR)	($) (VA – CR)	(%) (VC ÷ VA)	($) (VA – VP)	(%) (VP ÷ VP)	IDC (VA ÷ CR)	IDP (VA ÷ VP)
1.0 Plano pré-piloto	63.000	58.000	62.500	-4.500	-7,8	-5.000	-7,9	0,93	0,92
2.0 Listas de verificação	64.000	48.000	46.800	1.200	2,5	-16.000	-25,0	1,03	0,75
3.0 Currículo	23.000	20.000	23.500	-3.500	-17,5	-3.000	-13,0	0,85	0,87
4.0 Avaliação intermediária	68.000	68.000	72.500	-4.500	-6,6	0	0,0	0,94	1,00
5.0 Suporte à implementação	12.000	10.000	10.000	0	0,0	-2.000	-16,7	1,00	0,83
6.0 Manual de práticas	7.000	6.200	6.000	200	3,2	-800	-11,4	1,03	0,89
7.0 Plano de lançamento	20.000	13.500	18.100	-4.600	-34,1	-6.500	-32,5	0,075	0,68
Totais	**257.000**	**223.700**	**239.400**	**-15.700**	**-7,0**	**-33.300**	**-13,0**	**0,93**	**0,87**

Figura 9. Exemplo de relatório de desempenho tabular.

Comment [G757]: formatar

As previsões são atualizadas e refeitas com base nas informações sobre o desempenho do trabalho fornecidas conforme o projeto é executado. Essas informações se referem ao

desempenho passado do projeto que poderiam afetar o projeto no futuro, por exemplo, estimativa no término e estimativa para terminar.

3. Mudanças solicitadas

A análise do desempenho do projeto freqüentemente gera mudanças solicitadas em algum aspecto do projeto, conforme mostrada na seção 14.2.2.2.

Comment [G758]: deixar na proxima pagina

Comment [G759]:

- *Membros da equipe do projeto*: O grupo que está executando o trabalho do projeto.

Formatted: Font: Italic

- *Influenciadores:* Pessoas ou grupos que não estão diretamente relacionados à

pessoa na organização do cliente ou na organização executora, podem influenciar, positiva ou negativamente, no andamento do projeto.

Figura 10. Relação entre as partes interessadas.

Comment [G760]: formatar

A composição do processo é mostrada na Ffigura 11.

Comment [G761]: deixar um espaco entre texto e figura

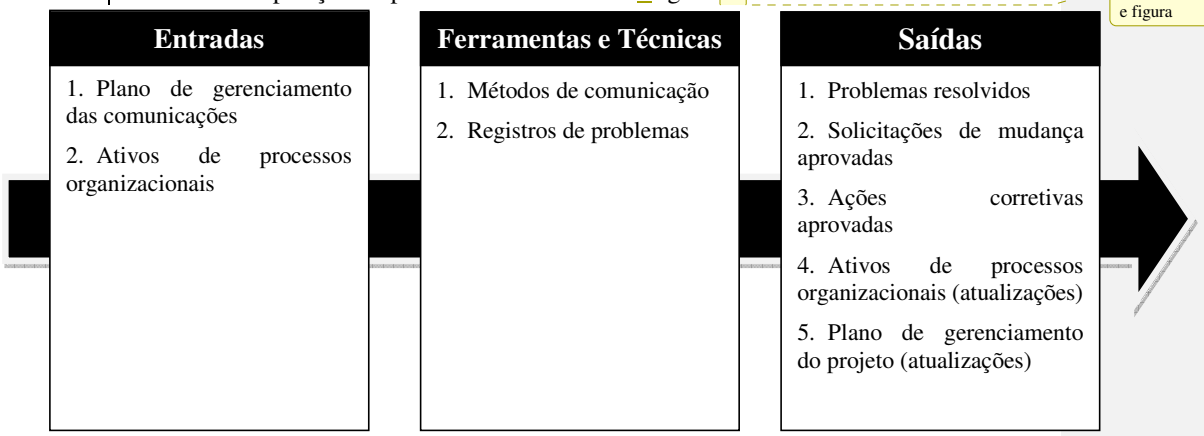


Figura 11. Gerenciar as partes interessadas.

Comment [G762]: formatar

Comment [G763]: espacamento muito grande

Comment [G764]: talvez seria interessante ter algo como consideracoes finais

O processo de desenvolvimento distribuído de software depende largamente da comunicação entre os envolvidos no projeto, seja de forma direta ou indireta. A Comunicação em DDS é caracterizada por diversos desafios como pessoas, processos, tecnologias, dispersão física, distância temporal e diferenças culturais. Esta nova maneira de se desenvolver software apresenta um grande impacto na forma como os

processo de comunicação se torna mais complexo, logo a escolha do meio de comunicação para a realização de determinadas tarefas exige cuidados.

Verzuh, E. (2000). MBA compacto, Gestão de Projetos. 3. ed. Rio de Janeiro: Campus.398 p.

Comment [A765]: Primeiro é Tópicos de Pesquisa e depois Sugestões de Leitura.

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 [A766]: Seria interessante neste espaço mostrar o objetivo do capítulo, antes de aborda o que o capítulo apresenta.

Comment [A767]: Acho que ficaria melhor deste jeito:
"*O capítulo aborda uma visão geral sobre medição de software, a sua importância, conceitos básicos,*

Além do que já foi dito no parágrafo anterior, boas práticas na implantação de

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.

Comment [A768]: 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 [A769]: 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 [A770]: Uma das causas que reflete este problema é uma gestão mal planejada e organizada

Comment [A771]: 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 [A772]: 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 [A773]: 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 [A774]: Ao invés de números acho que fica melhor colocar marcadores.

Comment [A775]: Na produtividade e na qualidade.

Comment [A776]: péssima

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, dando a clientes e à própria equipe uma certeza mais absoluta do que será necessário, em todos os sentidos, para a realização do mesmo.

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

Comment [A777]: acho que fica melhor **representar**

Comment [A778]: conceba

Comment [A779]: oferecer

Comment [A780]: seria melhor mudar essa expressão, pois ficou meio estranha no texto.

Comment [A781]: proporcionando

Comment [A782]: Acho que isso pode ser retirado.

Comment [A783]: Poderia retirar o dentro e ficar somente "…. nas atividades..."

Comment [A784]: Acho que poderia ficar assim: *"… na engenharia de software, por exemplo, …."*

Comment [A785]: Poderia retirar essa expressão e começar com o *"Na verdade..."*

Comment [A786]: Pode ser retirado

Comment [A787]: Feitas é no sentido de construídas ou de Realizadas?

Comment [A788]: Em geral fica melhor

Comment [A789]: Oferecerem

Comment [A790]: 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 [A791]: Melhor mudar essa expressão.

Comment [A792]: 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 [A793]: Definição de métricas

Comment [A794]:
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 [A795]: Acho que pode juntar com o paragrafo anteriror 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 tran... [1]

Comment [A796]: Acho que ficaria melhor assim:
... [2]

... [3]

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, poderíasse 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 erros encontrados por teste de unidade).

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:

Comment [A798]: 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.

Comment [A799]: 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).

Comment [A800]: Não tem mt haver com o que foi abordado anteriormente.

Comment [A801]: Colocar virgule depois de indicadores

Comment [A802]: 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 [A803]: Colocar virgule depois do permitindo assim.

Comment [A804]: Retirar a barra e colocar o "e".

Comment [A805]: Retirar o tudo isto.

Comment [A806]: Acho que fica melhor "retiradas".

Comment [A807]: Acho que ficaria melhor assim:
Tais medições também funcionam de forma

Comment [A808]: Excluir esse pedaço.

Comment [A809]: É melhor substituir por entre outros.

Comment [A810]: Substituir por: Segundo Pressman [Pressman 1995]

Nas próximas seções cada uma das categorias de métricas de software é estudada.

Comment [A811]: Substituit por : sera apresentada.

15.3.1 Métricas Técnicas

Comment [A812]: Vais acrescentar mais assunto neste tópico?

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 muito na complexidade do programa e na sua modularidade. Por sua vez, as métricas formadas depois da implementação do software dão uma indicação ao gerente e à equipe de 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 [A813]: Substituir por oferecem.

Comment [A814]: Excluir essas palavras.

Comment [A815]: Acho que é melhor assim. Uma atenção em especial

Guarizzo, Karina, (2008) "Métricas de Software", http://bibdig.poliseducacional.com.br/document/?view=184.

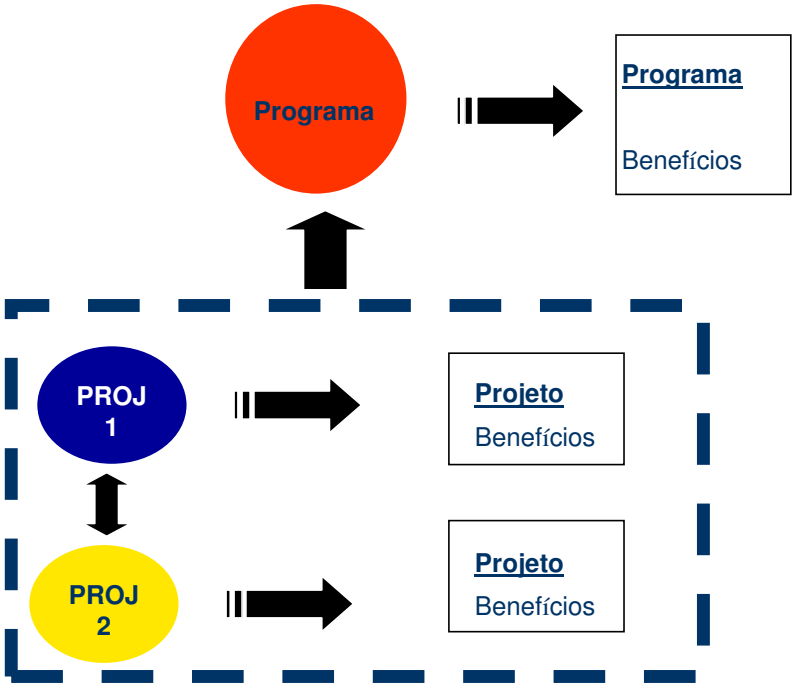
Comment [A816]: Organizar as referências, de acordo com o padrão estabelecido para o livro.

Capítulo

Programas

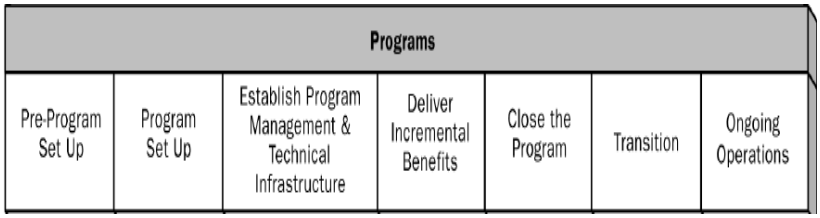
Segundo o IPMA(International Project Management Association), um programa consiste em um conjunto de propostas específicas e inter-relacionadas (projetos ou

outras tarefas adicionais), que em conjunto convergem para uma finalidade comum, segundo uma determinada estratégia abrangente.



Possuir um bom modelo de gestão capaz de envolver, comunicar e até mesmo influenciar os *stakeholders*, visando alcançar 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 a 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.


Programs
Pre-Program Set Up
Program Set Up
Establish Program Management & Technical Infrastructure
Deliver Incremental Benefits
Close the Program
Transition
Ongoing Operations

Nesta fase, inicia-se os projetos do programa; coordenar seus produtos para criar

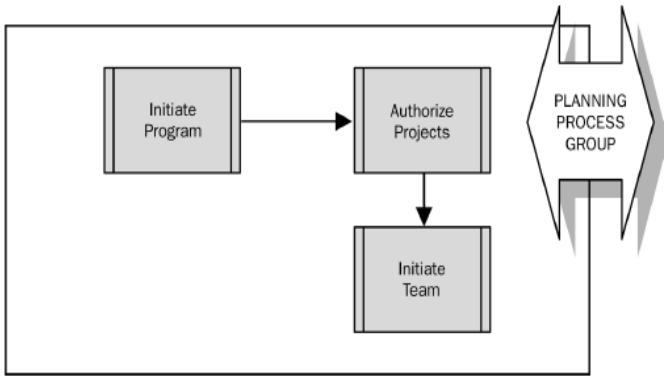
componentes, é iniciada; é iterativa; pode ter duração ilimitada; atividades são repetidas tantas vezes quantas necessárias e benefícios são atingidos cumulativamente. Principais atividades:

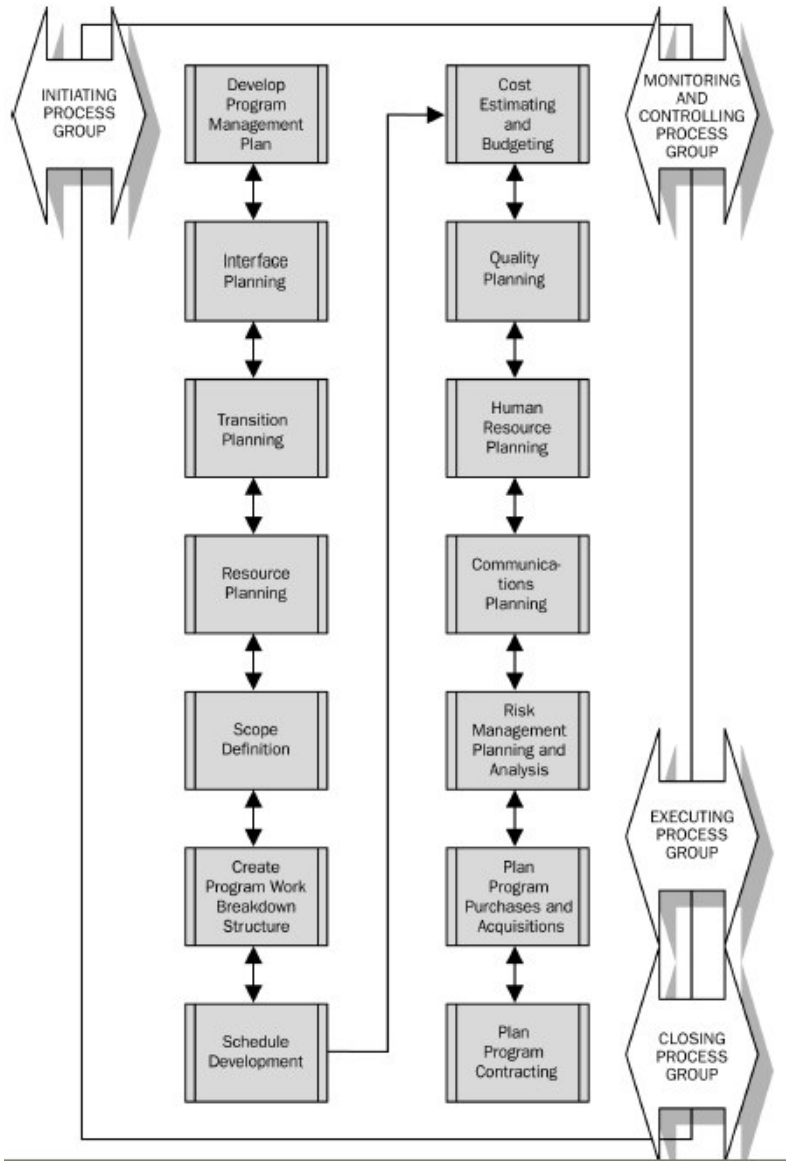
ENTRADA
• Declaração de escopo do programa • Critério de seleção dos projetos • Plano estratégico

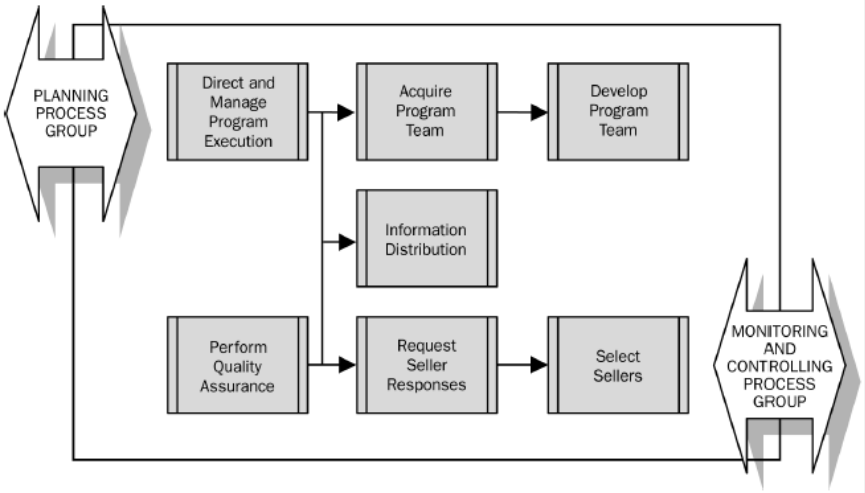
SAÍDA
• Requisitos de relatórios do Programa • *Project Charter* • Designação do gerente do projeto • Identificação do Patrocinador do projeto • Aprovação das reservas financeiras do projeto

ENTRADA
• Prática de recrutamento • Descrição de recursos

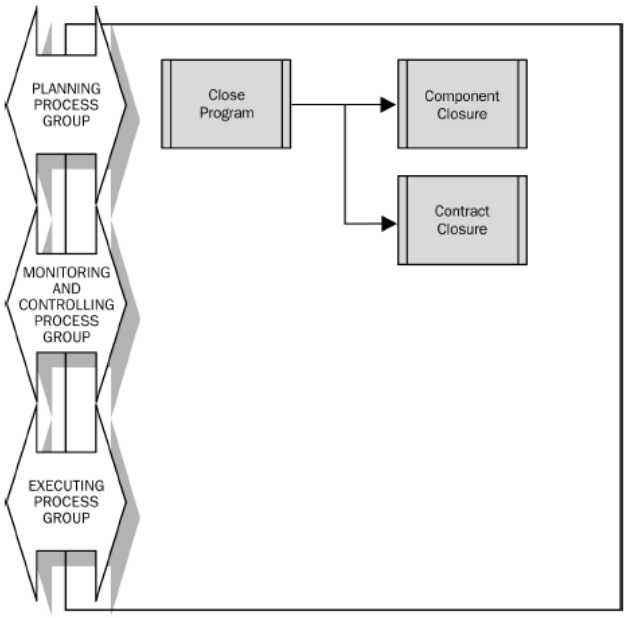
SAÍDA
• Núcleo do time do programa designado • Gerente do programa designado • Time do Programa


Initiate
Program
Authorize
Projects
Initiate
Team
PLANNING
PROCESS
GROUP


INITIATING PROCESS GROUP
Develop Program Management Plan
Interface Planning
Transition Planning
Resource Planning
Scope Definition
Create Program Work Breakdown Structure
Schedule Development
Cost Estimating and Budgeting
Quality Planning
Human Resource Planning
Communica-tions Planning
Risk Management Planning and Analysis
Plan Program Purchases and Acquisitions
Plan Program Contracting
MONITORING AND CONTROLLING PROCESS GROUP
EXECUTING PROCESS GROUP
CLOSING PROCESS GROUP


PLANNING PROCESS GROUP
Direct and Manage Program Execution
Acquire Program Team
Develop Program Team
Information Distribution
Perform Quality Assurance
Request Seller Responses
Select Sellers
MONITORING AND CONTROLLING PROCESS GROUP

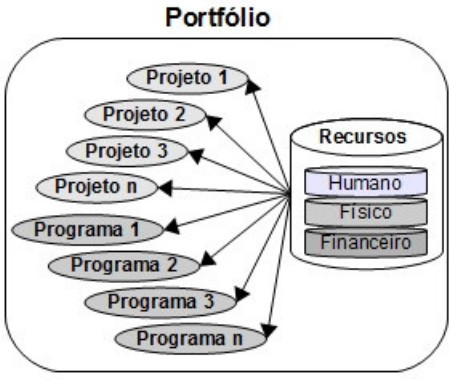
- Demonstrar que toda a documentação foi arquivada segundo o plano de


PLANNING PROCESS GROUP
MONITORING AND CONTROLLING PROCESS GROUP
EXECUTING PROCESS GROUP
Close Program
Component Closure
Contract Closure

[ADONAI 2008] Gestão de Programas e Múltiplos Projetos – Do conceito à

Capítulo

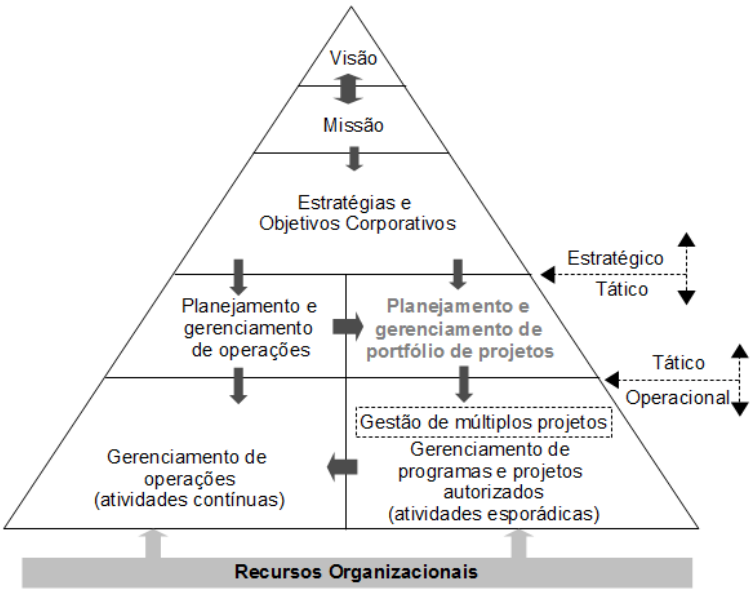
Definição de Portfólio



Estratégia Corporativa e Gestão de Portfólio

Tanto as ações operacionais, quanto as de projetos, devem ser consideradas na gestão de portfólio. As ações operacionais utilizam atividades recorrentes e processos de gestão operacional para facilitar a execução do planejamento de alto nível. Já os projetos utilizam processos de gerenciamento que permitam o planejamento e execução eficiente das atividades [PMI 2006]. Neste nível tático de gestão, uma das principais

recursos, com o mínimo esforço e em conformidade com valores e normas organizacionais.



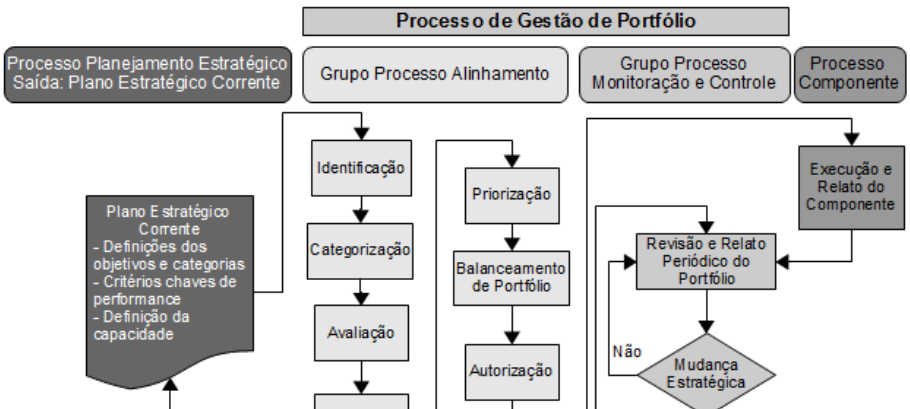
No caso da gestão de múltiplos projetos, que tem o foco muito mais operacional, devido ao fato de trabalhar em conjunto com a gestão dos projetos em execução, a

projetos estará constantemente analisando a alocação dos recursos e re-distribuindo os disponíveis entre seus projetos. Também poderá fazer parte do conjunto de atividades da gestão de múltiplos projetos o auxilio na resolução de conflitos no âmbito de cada projeto, apoio técnico em metodologia de gestão de projetos, além do acompanhamento gerencial para que os projetos alcancem as metas estabelecidas e obtenham sucesso em relação aos seus requisitos.

Gerente de Portfólio

Os modelos e padrões de gestão de portfólio são propostas de processos de trabalho, criados para sistematizar a dinâmica decisória do portfólio [Blomquist and Muller

execução dos componentes que realmente estejam alinhados com as estratégias da organização e que proporcionem maior valor agregado.



informações são levantadas para cada componente do portfólio e podem ser qualitativas ou quantitativas, vindo das mais variadas fontes da organização. Deve-se ter o cuidado necessário para não comprometer a precisão das informações durante a atividade de levantamento. Gráficos, diagramas, documentos e recomendações são gerados para apoiar o processo de seleção.

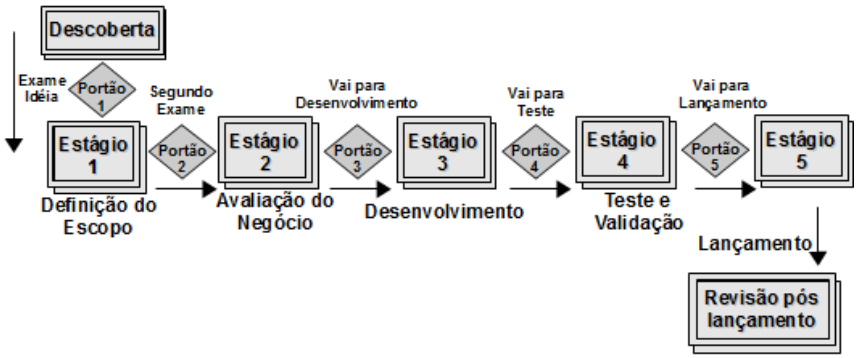
O objetivo deste processo é proporcionar a combinação de componentes com maior

benefícios da gestão do portfólio, que são a capacidade de planejar e alocar recursos(financeiro, físicos e humanos) alinhado com as orientações estratégicas e a capacidade de maximizar o retorno da portfólio dentro do que foi predefinido em termos de risco aceitável pela organização.

O principal objetivo das revisões é assegurar que o portfólio contém apenas componentes que realmente estejam alinhados como os objetivos estratégicos da organização. Para garantir isso, os componentes devem ser periodicamente adicionados, realinhados, ou até mesmo removidos, com base no seu desempenho e alinhamento com as estratégias definidas, a fim de assegurar uma gestão eficaz do portfólio.

A entrada de cada estágio é um ponto de decisão, de modo que estes pontos

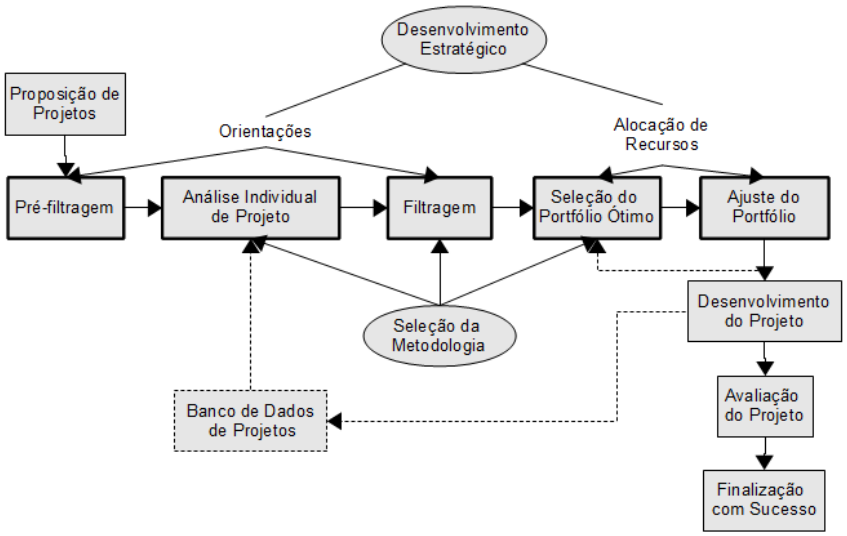
tomada de decisão que englobam as seguintes saídas: Inicia, Continua, Aborta, Suspende ou Recicla.



• **Estágio 2 – Avaliação do Negócio**

Neste momento será realizada uma avaliação criteriosa do projeto onde é revisado todo o seu desempenho e seus pontos fortes e fracos. Todas estas informações serão

desenvolvimento de novos projetos, aumentando assim a qualidade dos projetos da organização. Esta etapa marca o fim do projeto.



Nesta fase, que utiliza informações produzidas no Desenvolvimento Estratégico, as

portfólio estejam alinhados com os objetivos estratégicos da organização. Nesta fase também será realizada uma análise de viabilidade e estimativas de parâmetros necessários para avaliar cada projeto, bem como as metas para cada parâmetro, o que será uma fonte de informações para outras fases.

Estudo de Caso: Gestão de Portfólio de Projetos no SERPRO

acompanhamento dos projetos. Durante os ciclos cada gerente de projeto emiti 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.

O processo contempla a análise de cenários, com base em métricas, para tomadas de decisões que proporcionem valor agregado à organização, buscando assim

especificamente no momento da realização do alinhamento do Portfólio às estratégias da organização.

empresa de desenvolvimento de software na escolha do portfólio de projetos mais adequado à realidade da organização. O modelo proposto possibilita a criação de uma ligação entre os projetos e a estratégia da organização e auxiliará na adoção uma visão de longo prazo.

[Reyck et al 2005] Bert De Reyck, Yael Grushka-Cockayne, Martin Lockett, Sergio

management on information technology projects. International Journal of Project Management, Volume 23, Issue 7, October 2005, Pages 524-537

[SERPRO 2009] http://www.serpro.gov.br

O Escritório de Projetos ou *Project Management Office* (PMO) tem sido uma das estruturas de Gerenciamento de Projetos mais buscadas atualmente pelas organizações. Tal procura dar-se 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) [1].

Por se tratar de uma estrutura relativamente recente, existe ainda uma grande discussão a respeito dos seus papéis, responsabilidades e formas de implementação. Estudos têm buscado levantar sua importância estratégica através principalmente da demonstração de que o PMO, quando implantado de forma adequada, pode se mostrar como um excelente agregador as estruturas organizacionais.

Comment [Carlos Al817]: Só usar tabulação a partir do segundo parágrafo de cada seção

Comment [Carlos Al818]: Trocar a palavra por outra melhor

Comment [Carlos Al819]: O padrão de referência é [Autor Ano]

Comment [Carlos Al820]: pode-se??

– Papéis e funções

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 Al821]: Sugiro a criação de sub-tópicos para detalhar melhor as funções e grupos nesta seção

Comment [Carlos Al822]: Só usar tabulação a partir do segundo parágrafo de cada seção

Comment [Carlos Al823]: Retirar os enters entre os parágrafos.

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) [6] são: “orientar e dar suporte aos gerentes de projetos permitindo à empresa desenvolver seus projetos da forma mais eficiente e eficaz possível”.

Comment [Carlos Al824]: Colocar vírgula

O apoio na comunicação eficiente entre os gerentes de projeto e a alta administração, e por natural conseqüência ao planejamento estratégico organizacional, tem sido outro objetivo bastante destacado entre os PMOs.

Comment [Carlos Al825]: Não ficou clara a relação de apoio a comunicação em relação ao PEO

Algumas organizações têm utilizado o PMO para uniformizar processos, práticas e operações de Gerenciamento de Projetos. Este é outro objetivo também bastante difundido na atualidade. Os processos uniformizados devem acarretar a resultados mais sólidos e repetíveis, como também a uma maior perspectiva de sucesso nos projetos executados. A idéia aqui é atenuar o atraso dos projetos, identificar e abrandar os riscos, além de promover uma estimativa de custos mais realista.

Comment [Carlos Al826]: Tirar o “a” antes de resultados.

- Tornar possível a execução dos projetos de forma que os mesmos fiquem alinhados com os objetivos da alta direção. A disposição do PMO possibilita a concentração das informações e condução dos projetos, facilitando a disposição dos objetivos dos projetos com os objetivos da organização.
- Recolher e analisar informações sobre o desempenho dos projetos, com a finalidade de identificar deficiências e melhores práticas, buscando a resolução dos problemas e a propagação das melhores práticas.

Comment [Carlos Al827]: Não ficou claro

de projetos presentes na organização e encontrando possíveis deficiências. Programas de capacitação e disseminação de conhecimentos e habilidades na área são uma grande preocupação neste caso.

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) [8].

– **Classificações dos PMOs**

Kerzner (Kerzner, 2005) [9] classifica os PMOs em três tipos: Escritório de Projetos Funcional, Escritório de Projetos de Grupo de Clientes e Escritório de Projetos Corporativo. 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 Al829]: Sugiro a criação de sub-tópicos para detalhar melhor as funções e grupos nesta seção

Comment [Carlos Al830]: Acredito que se trate de tipos, não classificações

Comment [Carlos Al831]: Poderia se utilizar de tópicos para melhorar o entendimento

Uma segunda classificação é apresentada por (DINSMORE 2003) [6]. Este tipo de classificação é apresentado em cinco tipos distintos definidos da seguinte forma:

Comment [Carlos Al832]: Poderia se utilizar de tópicos

A figura abaixo apresenta o posicionamento de uma APT sobre uma estrutura organizacional baseada nos modelos propostos por (VARGAS - 2003)[10] e (DINSMORE - 2003)[6].

Comment [Carlos Al833]: Citar a figura pela referência. Ex: De acordo com a Figura 1, ...

Figura 1 - APT: Equipe Autônoma de Projeto – adaptado por Gerhard (GERHARD, 2004) de Dinsmore (DINSMORE, 2003) e Vargas (VARGAS, 2003).

Comment [Carlos AI834]: Resolução e legibilidade da figura está ruim

No formato Escritório de Suporte de Projetos (PSO - *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 AI835]: Sigla já citada anteriormente.

Vargas (VARGAS, 2003) afirma ainda que um PSO pode ser departamental ou corporativo, variando apenas em sua colocação dentro da estrutura organizacional da empresa. A figura 2 apresenta um organograma por departamento:

Comment [Carlos AI836]: No singular é com a 1ª letra maiúscula; no plural tudo minúsculo.

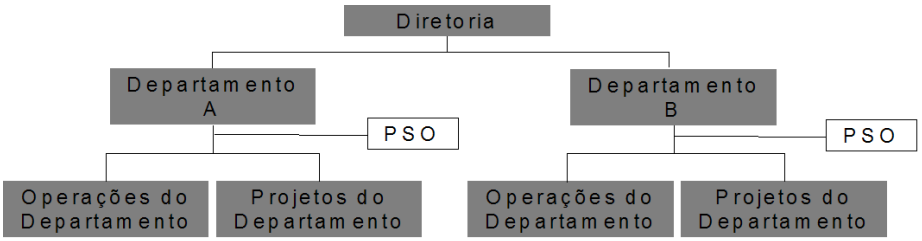


Figura 2 – PSO por departamento segundo Vargas (VARGAS, 2003).

Comment [Carlos Al837]: Resolução e legibilidade da figura está ruim

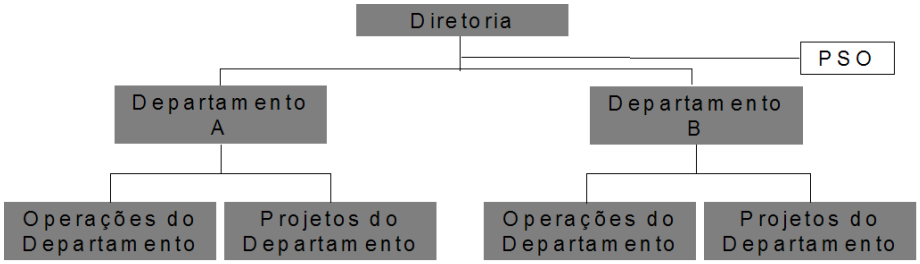


Figura 3 – PSO Corporativo segundo Vargas (VARGAS, 2003).

Comment [Carlos Al838]: Resolução e legibilidade da figura está ruim

Centro de Excelência em Gestão de Projetos *(PMCOE – 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 é 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 Al839]: Sigla já citada anteriormente.

Conforme Kerzner (KERZNER, 2005) o Centro de Excelência em Gestão de Projetos assume como responsabilidade maior fazer com que se municiem informações às partes diretamente interessadas ao invés de comandar projetos. Outro fator interessante

este tipo de estrutura, porém se pode alcançar resultados fantásticos mesmo com recursos em tempo parcial. Dinsmore (DINSMORE, 2003) complementa ainda que a outra missão do PMCOE é a de alastrar a cultura de gerenciamento de projetos na organização, sendo responsável por manter as metodologias.

O Escritório de Gerência de Programas (PrgMO – *Program Management Office*) por sua vez apresenta como característica a gerencia 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 Al840]: Sigla já citada anteriormente.

Figura 3 – PrgMO Corporativo segundo Vargas (2003).

Comment [Carlos AI841]: Resolução e legibilidade da figura está ruim

Por fim, o *Chief Project Officer* (CPO) é um cargo que está no cume da pirâmide organizacional. Este tipo de estrutura acaba sendo raramente utilizada. Segundo esta estrutura o administrador e sua equipe direta têm autonomia sobre os projetos estratégicos, incluindo a tomada de decisão sobre negócios que implicam no desenvolvimento de novos projetos, o planejamento estratégico, a escolha de prioridades e nas transações de recursos para os projetos, a visão geral sobre a implementação e desenvolvimento do sistema de gestão integrada, revisão periódica e decisão sobre interrupção de projetos, além do gerenciamento de *stakeholders* de alto nível.

Comment [Carlos AI842]: Trocar por partes interessadas

– Todas as funções de um EGP de Nível 1;

Comment [Carlos Al843]: Sigla não especificada anteriomente.

- Quais são os resultados esperados do PMO; Exemplo: foco no repasse de informação para a alta gestão, padronização de procedimentos ou até mesmo gestão de *stakeholders*;

- Quais as características da organização que podem promover ou dificultar a implementação do PMO.

- **Boas práticas na Implantação de PMOs**

Comment [Carlos Al844]: Trocar por partes interessadas

Comment [Carlos Al845]: Seria interessante colocar um pequeno processo que mostrasse passo a passo as boas práticas

Kerzner acredita que se deva enfatizar alguns fatores na implantação de um Escritório de Projetos:

Comment [Carlos Al846]: Tive a impressão de estar lendo as principais atribuições/atividades de um PMO.

Para atender estes itens, o PMO deve explorar junto à alta diretoria para sempre manter-se alinhado ao planejamento estratégico da empresa, realizar auditorias freqüentes nos projetos da empresa, medir periodicamente o grau de maturidade em Gerenciamento de Projetos da corporação, rever o procedimento aplicado nos projetos, manter organizado e atualizado o conjunto de recursos, prover treinamentos e primar pela disseminação da cultura de Gerenciamento de Projetos dentro da organização.

– **Caso de sucesso na implantação de um PMO**

Comment [Carlos Al847]: Sugiro colocar o nome do SERPRO neste tópico. Ex: Estudo de Caso: Implantação do Escritório de Projetos do SERPRO.

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 Educacão.

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.

Neste período a empresa conseguiu ampliar seu faturamento, com o aumento dos investimentos, podendo assim atender a alguns dos objetivos firmados com o governo. Dar retorno ao próprio governo, permitir seu reinvestimento para atualização

objetivos são muito próximos aos de uma empresa privada. Sua atividade fim é o desenvolvimento de software, possuindo clientes atendidos por diversas unidades espalhadas pelo território brasileiro e demandas das mais variadas complexidades, algumas simples, outras complexas.

– **Estratégia da Implantação**

Comment [Carlos Al848]: Sugiro usar até 3 níveis. Após isso usar marcadores.

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 aculturamento em gerenciamento de projetos deveria ser provido a toda a Empresa, por meio de curso E@D e o esforço e

melhoria e da viabilidade de integração com as demais ferramentas existentes no Serpro.

Passa a acontecer planejamento anual do processo, onde, baseando-se no planejamento estratégico da empresa e em orientações táticas da Diretoria, são

sua vez, ficam subordinados a um projeto maior que responde pelo escopo total para aquele ano. Este projeto é gerido pelo coordenador do GT-SGPS e seus subprojetos, geridos e executados pelos demais membros do grupo de trabalho.

- Revisar estratégia de capacitação de pessoas de forma a otimizar os resultados do incentivo à certificação PMP e implantar treinamentos avançados em práticas críticas como construção de EAP e cronogramas.

Comment [Carlos AI849]: Sigla não especificada anteriomente.

- **Tópicos de Pesquisa**

Comment [Carlos AI850]: Citar as fontes das pesquisas

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:

- Desenvolvimento de Estratégias para Implantação de PMOs;

Comment [Carlos AI851]: Acrescentar uma breve explicação sobre cada tópico.

– 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 AI852]: Resolução e legibilidade da figura está ruim

– 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 AI853]: Resolução e legibilidade da figura está ruim

[15] GERHARD, Eduardo Causas e Conseqüências da implantação de um PMO – Project Management Office, São Leopoldo, novembro de 2004. Disponível em: < http://www.unisinos.br/inf/images/stories/Inf/22tc_eduardo_gerhard.pdf/>. Acesso em 18/09/2009.

[19] Implantação do Gerenciamento de Projetos no SERPRO. Iran M. Porto Junior, João V. de Melo Neto. Artigo público. 2008.

[20] 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 P2009.

Comment [Carlos Al854]: Acrescentar um ponto entre o autor e a fonte

Comment [Carlos Al855]: Colocar o local da publicação.

Comment [Carlos Al856]: Colocar o autor da monografia.

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 e para isso é importante que o significado dos mesmos seja conhecido para garantir o pleno entendimento de todo o contexto.

Comment [0857]: Parágrafo muito extenso. Sugestão: Quebrar em dois.

é a aplicação de conhecimentos, habilidades, ferramentas e técnicas nas atividades do projeto a fim de atender os requisitos do projeto.

Comment [0858]: Estes conceitos não são apresentados em capítulos anteriores?
Seria interessante fazer um link caso isto seja verdade?

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.

Comment [0859]: Para isso,

Na próxima seção serão descritos cinco modelos de maturidade que estão entre os mais conhecidos e utilizados atualmente.

Comment [0860]: Listar quais serão apresentados.

O estudo da maturidade em gerenciamento, assim como o estudo do próprio gerenciamento de projetos, é assunto que vem sendo discutido há pouco tempo, mas tem ocupado lugar de destaque[Leal 2008].

Comment [0861]: Sugestão: recentemente

A proposta do OPM3 (Organizational Project Management Maturity Model) é se tornar um amplo modelo de maturidade que seja endossado e reconhecido mundialmente como um padrão para desenvolver e avaliar as capacidades de gerência de projetos em qualquer categoria de organização [PMI 2003].

Comment [0862]: Itálico

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.

Comment [0863]: Palavras em Inglês em Itálico

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.

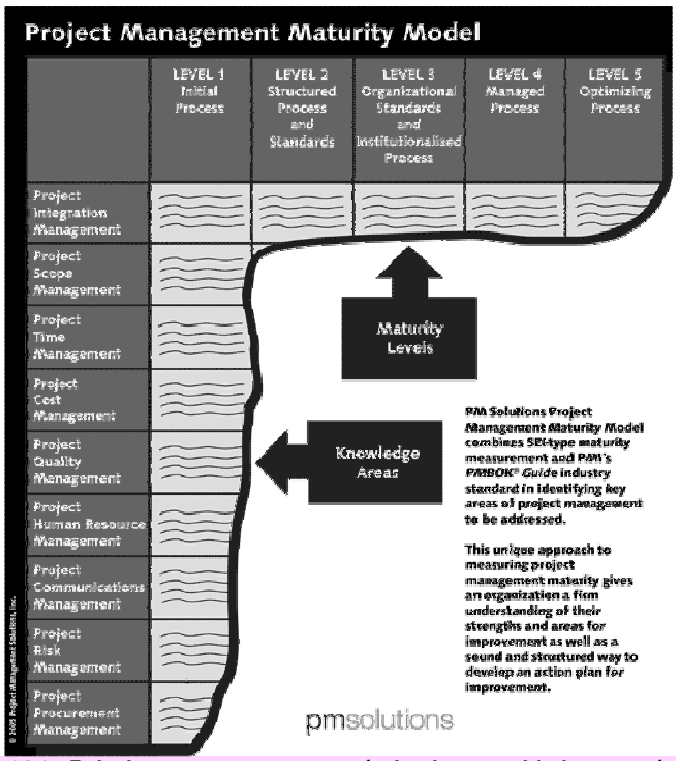


Figura 14.1. Relacionamento entre os níveis de maturidade e as áreas de conhecimento do PMMM [Crawford 2002]

Comment [0864]: Figura com termos em inglês.

A Figura 14.1 demonstra a proposta da PM Solutions que foi criar um modelo de maturidade incorporando os 5 níveis de maturidade evolucionais existentes no Capability Maturity Model (CMM) desenvolvido pelo Software Engineering Institute (SEI) e as nove áreas de conhecimento do PMBOK.

Comment [0865]: Itálico

O modelo de maturidade P3M3 (Portfólio, Programme and Project Management Maturity Model) foi desenvolvido pela *OGC - Office of Government Commerce* e

Comment [0866]: Itálico

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 P3M3 contempla em seu conteúdo a avaliação da maturidade no gerenciamento de projetos, programas e portfólios, e é tido como uma evolução de um modelo de maturidade anterior desenvolvido pela OGC, o P1M3 – Project Management Maturity Model, que possuía como escopo o gerenciamento de projetos. A partir deste modelo, a OGC desenvolveu os modelos P2M3 - Programme and Project Management Maturity Model, que contempla projetos e portfólios e o P2MM - PRINCE2 Maturity Model que contempla a maturidade no uso da metodologia PRINCE2 em gerenciamento de projetos.

Comment [0867]: Itálico

Comment [0868]: Itálico

	Gerência e comunicação com os stakeholders

Comment [0869]: Itálico

O modelo KPMMM (Kerzner Project Management Maturity Model), desenvolvido pelo Dr. Harold Kerzner, em 1998, o modelo define o estágio atual, o planejamento e ações para implementação e desenvolvimento gradual da gerência de projetos [Kerzner 2003].

Comment [0870]: Itálico

Figura 14.2. O conhecimento dirige a avaliação, que, por sua vez, dirige a melhoria [PMI 2003]

Comment [0871]: Figura com termos em inglês.

- 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 [0872]: Inserir "."

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 de ações e consequente evolução em seus estágios de melhoria do processo. Cada domínio se comunica entre si por meio de ligações entre seus grupos de processos [Harrisson 2006]. A Figura 14.3 demonstra o relacionamento entre os domínios, os estágios e os grupos de processos especificados no modelo.

Figura 14.3. Representação do modelo OPM3 em termos dos domínios, estágios de melhoria do processo e grupos de processos[PMI 2003].

Comment [0873]: Sugestõa: caixas laterais com a mesma "textura" das caixas inferiores para facilitar visualização dos nomes.

O modelo evidencia a importância de utilizar um instrumento de avaliação, constituído por um questionário, através do qual o usuário analisa e responde sobre a presença ou não de processos formais associados ao ciclo de vida do gerenciamento de projetos [PMI 2005].

Comment [0874]: Ao final da apresentação dos modelos, caso seja possível e achares interessante, realizar um comparativo.

Este questionário é composto por 151 perguntas abrangendo os domínios de Projetos

domínio que avaliavam (70 perguntas para Projeto, 38 perguntas para Programas e 43 perguntas para Portfólio).

Figura 14.4. Relacionamento entre as dimensões “estágio de melhoria de processos” e “domínios” [PMI 2003].

Comment [0875]: Figuras com termos em inglês

Figura 14.5. Passos para implantação do Modelo OPM3 [PMI 2003].

Comment [0876]: Figuras com termos em inglês

Os cinco níveis de maturidade existentes no MMGP setorial são os seguintes: inicial, conhecido ou linguagem comum, padronizado, gerenciado e otimizado. O relacionamento existente entre as dimensões e os níveis de maturidade será mostrado na tabela a seguir [Prado 2004]

Comment [0877]: Retirar espaços

Avaliação Final = (100 + somatório total de pontos das perguntas) / 100

Comment [0878]: colocar como figura

Figura 14.6. Níveis de Maturidade do Modelo KPMMM [KERZNER 2002]

Comment [0879]: Melhorar labels das figuras

- Nível 4 – Benchmarking

Comment [0880]: Itálico

- Maturidade

Comment [0881]: inserir ; para cada item

- Criar um programa de treinamento contínuo em gerenciamento de

gerenciamento de projetos possam ser mantidos e melhorados a longo prazo.

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 identificado que o modelo mais adequado para esta pesquisa seria o Modelo de Maturidade em Gerenciamento de Projetos – MMGP proposto por Darci Prado [Prado 2004].

Comment [0882]: Se tivesse como colocar o porque seria interessante.

Este questionário, que pode ser visto na íntegra em [Prado 2004] ou [Carneiro 2007], foi aplicado nas empresas juniores brasileiras através do envio de um convite eletrônico explicando o propósito da pesquisa e contendo o link para o site onde o

Comment [0883]: o qual

representante da empresa respondente, sendo este necessariamente uma pessoa que possuísse plenos conhecimentos acerca dos processos praticados no gerenciamento de projetos e atuasse em um cargo estratégico, ou seja, fizesse parte da diretoria executiva da empresa júnior.

14.6.3. Perfil dos participantes

Comment [0884]: Não seria o caso de apresentar primeiro o perfil dos participantes ?

As empresas juniores – EJ's respondentes estão localizadas em 3 das 5 regiões brasileiras como descrito no gráfico abaixo, a ausência de participação das regiões norte

participar da pesquisa, porém nenhuma de suas EJ's enviou o questionário respondido. Sendo assim foi difícil estabelecer contato com estas regiões inviabilizando participação destas na pesquisa.

Outra segmentação importante para identificar o perfil dos participantes é relativa à área de atuação das EJ's, que pode ser vista no gráfico a seguir:

Comment [0885]: Foi descrito o que é EJ antes?

A maturidade média das EJ's brasileiras que responderam a pesquisa é de 2,41. O valor é praticamente idêntico ao obtido na Pesquisa Archibald & Prado 2006 – Maturidade em Gerenciamento de Projetos [Prado, Archibald 2006], utilizando também o Modelo Prado-MMGP setorial e o Modelo de Categorias de Archibald, quando foi obtido o índice médio de 2,42. Desta pesquisa participaram 258 empresas de 4 tipos de organizações, iniciativa privada, governo (administração direta), governo (administração indireta) e terceiro setor. Essa comparação entre as duas pesquisas confia credibilidade a ambas visto que a maturidade de um país tem alguma estabilidade no tempo. O valor médio 2,41 possui a seguinte distribuição (vide figura abaixo)

Comment [0886]: Colocar :

A seguir iremos detalhar como se distribui o Nível de Maturidade das EJ's Brasileiras em relação às regiões, estados e área de atuação

Comment [0887]: Inserir pontuação adequada

Abaixo segue uma tabela com os níveis de maturidade mínimos médios e máximos atingidos por localização geográfica

Comment [0888]: Inserir pontuação adequada

Podemos concluir a partir dos dados acima que o estado de Minas Gerais é aquele que atingiu o maior nível de maturidade e possui também a maior variação entre o nível mínimo e o máximo. O estado da Paraíba assim como o do Paraná são aqueles

uniformidade. O estado da Bahia possui a empresa júnior com o menor nível de maturidade em toda a pesquisa.

No gráfico abaixo se observa que os estados do RJ e PR são os que possuem os maiores PANM 2, já em relação ao PANM 3 os estados que aparecem com maior aderência são BA e MG, enquanto que no PANM 4 a situação se inverte, o estado de MG lidera seguido pela BA, por fim no nível mais elevado de maturidade MG mantém a liderança só que desta vez seguido pelo RJ. Algumas curiosidades podem ser encontradas neste gráfico como: o estado da Bahia possuir o segundo pior PANM 2 e o maior PANM 3, a similaridade muito alta entre os PANM 3 entre todos os estados com exceção da PB que aparece em último em todos os PANM e os estados de MG e da BA possuírem PANM 3 maiores que os PANM 2 respectivamente.

Comment [0889]: Seria interessante descrever as siglas antes de referenciá-las?

Exemplo: RJ – Rio de Janeiro?

Comment [0890]: Não entendi muito bem o que é?

	MMGP DARCI PRADO
	Formato escalar, em níveis de 1 a 5 onde é possível a organização ter uma única nota geral através de uma fórmula de média aritmética. Mas, além disto, o modelo dá uma nota em percentual do grau de aderência da organização em cada nível. Assim, ele admite que a organização possa ter ao mesmo tempo características de vários níveis.
	O MMGP aborda a maturidade em gerenciamento de projetos sob o ponto de vista de 5 dimensões: conhecimentos de gerenciamento, metodologias, relacionamentos humanos, estrutura organizacional e alinhamento com os negócios. Além disso, ele se encaixa no sistema de gestão da consultoria do INDG e possui duas variantes: o MMGP setorial e corporativo.
	Como se trata de um modelo proprietário, este item é definido caso-a-caso entre a consultoria e a empresa. Mas, devido ao formato do questionário, aonde as respostas da letra "a" conduzem a mais pontos que as da letra "d" a nenhum ponto, a empresa já tem esboçado um caminho para sua melhoria.

Figura 1 – Modelos de Avaliação da Maturidade – Critérios [Oliveira 2006]

	PMMM HAROLD KERZNER	CMM	OPM3	MMGP DARCI PRADO
Disponibilidade	Modelo proprietário, desenvolvido pela experiência de seu autor. Encontra-se disponível através do livro *KERZNER H. Strategic planning for project office management using a project management maturity model. New York, John Wiley & Sons, 2001;* consultoria do ILL - International Institute for Learning e on-line mediante pagamento no site do ILL	Criado num ambiente acadêmico pela SEI - Software *Engineering Institute* e desenvolvido posteriormente por centenas de profissionais e empresas, encontra-se disponível através de diversos livros e consultorias específicas.	Criado por um processo voluntariado está disponível em livro (através do documento ***Knowledgement Foundation***) e em CD ROM perante pagamento.	Modelo proprietário, desenvolvido pela experiência de seu autor. Encontra-se disponível através de uma série de livros do autor e on-line gratuitamente no *site* www.maturityresearch.com
Formato do questionário	183 questões de vários tipos de dívidas desigualmente em 5 níveis de maturidade somando 800 pontos	Existem vários formatos de avaliação sendo que o SEI reconhecia até 2004, seis formatos de avaliação.	151 perguntas com respostas binárias (sim ou não) num ambiente de 4 dimensões: esferas de aplicação estágios de melhoria, avanço das capacitações conduzidas pelas melhores práticas e processos de gerenciamento de projetos.	40 perguntas de múltipla escolha., cada nível de 2 a 5 com 10 perguntas cada. O nível 1 é considerado inicial *ad-hoc*. Devido á forma de concepção das respostas no questionário, ele exigirá das organizações humildade e reconhecimento de suas fraquezas para que a pontuação seja o melhor reflexo da maturidade atual.

Comment [0891]: Criar tabela ao invés de figura e ver uma melhor forma de disposição

Figura 14.8. Modelos de Avaliação da Maturidade – Critérios (cont.) [Oliveira 2006]

Comment [0892]: Criar tabela ao invés de figura e ver uma melhor forma de disposição

Management Maturity Model (OPM3) Foundation. Newton Square Pennsylvania, de autoria do PMI - Project Management Institute.

- Caso haja interesse em obter mais informações sobre o P3M3 – Portfólio, Programme and Project Management Maturity Model pode ser lida a versão 2.0 do modelo que está disponível para consulta pública em: <http://www.p3m3-officialsite.com/nmsruntime/saveasdialog.asp?lID=322&sID=90>

Comment [0893]: Itálico

- Para obter maiores detalhes sobre a Avaliação da Maturidade em Gestão de Projetos nas Empresas Juniores do Brasil deve-se ler o trabalho de graduação: [CARNEIRO 2007]. Disponível através do link a seguir: <http://www.cin.ufpe.br/~tg/2007-2/desc.pdf >

Comment [0894]: Acessível quando?

<http://www.maturityresearch.com/2006/downloads/RelatorioFinal_Completo_MPCM_2006.pdf> Acesso em: 15 dez. 2007

Chapter

23.

Governança em TIC

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.

23.1 Gestão em TIC

Com o crescimento populacional, a globalização e o desenvolvimento do capitalismo no século XX, surgem novas necessidades para o ser humano. A quantidade de dados e de informações para serem armazenadas e computadas atinge um volume incalculável. A informática surge neste contexto: superar a necessidade do ser humano de registrar e de manipular dados em grandes quantidades com precisão e rapidez [NORTON 1997].

Apesar de bastante presente atualmente, a definição de informática não é tão simples, pois envolve conceitos abstratos. O termo informática foi criado em 1957, pelo cientista Karl Steinbuch, em um artigo que trata do processamento automático da informação [STEINBUCH 1957]. A partir daí, o termo foi traduzido para o francês, espanhol e português, sendo mais usado em idiomas latinos. A informática refere-se ao conjunto das Ciências da Computação e da Informação que, por sua vez, dedicam-se ao estudo da informação desde a sua gênese até o processo de transformação de dados em informação, e desta, em conhecimento.

Na década seguinte, em 1980, ocorreram mudanças tecnológicas no ambiente de escritório e a popularização dos microcomputadores (Personal Computers - PCs). Estas

Comment [M895]: Acho que deveria ser "Capítulo" e não Chapter e a fonte deve ser tamanho 18

Formatted: Bullets and Numbering

Comment [M896]: Texto deveria estar alinhado a esquerda e com fonte tamanho 20

Comment [M897]: Verificar se o texto que introduz o capítulo vai ficar escrito em "itálico" ou "normal"

Comment [M898]: Fonte deve ser no tamanho 14 para seções

Formatted: Bullets and Numbering

Comment [M899]: Não deve existir espaçamento vertical entre um parágrafo e outro.

Comment [M900]: Na formatação do parágrafo deve-se incluir um espaçamento de "6" antes

Informação - TI" passou a ser mais frequentemente empregado, ampliando o contexto do que era conhecido como informática. Este período ficou conhecido como a **Era da Inovação e da Vantagem Competitiva** [FOINA 2001].

Para se ter uma noção do valor financeiro originário de problemas nos serviços de TIC, basta analisar o quanto uma organização depende de tais serviços para consecução dos seus negócios, através da estimativa dos prejuízos gerados em perda de receita, por hora, no caso de interrupção em um dos seus servicos de TIC. Isso pode

suficientemente significativo para o tema merecer uma atenção especial. Uma idéia deste impacto pode ser analisada na Tabela 23.1 abaixo.

Tabela 23.1– Custo horário médio de interrupção de Serviços de TIC por natureza do negócio. FONTE: [MAGALHÃES 2007].

Comment [M901]: Texto descritivo da legenda não deve ser em negrito

Comment [M902]: Fonte tamanho 12

Formatted: Bullets and Numbering

23.1.1 Relevância e Evolução do Papel da TIC nas Organizações

Cada vez mais, no ambiente corporativo, as organizações vêm tomando ciência, de forma progressiva, da crescente importância que a Tecnologia da Informação e Comunicação – TIC está assumindo como fator impulsionador e catalisador dos aspectos de mudança, renovação e concretização do ciclo dos seus negócios. Da mesma

desdobramentos estão se tornando fatores estratégicos no aumento de sua competitividade mercadológica e na realização de sua missão institucional [EUROCOM 2006].

Figura 23.1– Evolução da Gestão dos Departamentos de TIC.
Fonte: Adaptado de [FERNÁNDEZ 2008].

Comment [M903]: Fonte tamanho 12

Comment [M904]: A "Fonte:" deve estar na mesma linha que a legenda quando possível

Comment [M905]: Todo o texto da legenda com exceção de "Figura 23.x" não deve estar em negrito

- Tudo o que precisa fazer é ler todos os livros Governança de TIC;

Comment [M906]: Acho que ficaria melhor se fosse escrito assim: "Tudo o que precisa ser feito é..."

Sem esquecer o fato que as organizações que optam por implantar Governança de TIC não estão imunes à maioria dos problemas enfrentados por seus gestores na

suas práticas no corpo de conhecimento de Gerenciamento de Projetos disponível, dentre os quais podemos destacar o PMBOK [PMBOK 2008].

5. A liderança do CIO[19]: 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 CEO[20] 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.

Comment [M907]: Os itens devem utilizar os marcadores do tipo "bullet" bolinha preenchida

Formatted: Bullets and Numbering

23.1.2Da Gestão à Governança em TIC

À medida que as organizações começaram a reconhecer a sua dependência crescente na TIC para conseguirem satisfazer os objetivos do negócio, caminhando no encontro às necessidades da organização, muitos autores determinaram como fundamental a

Comment [M908]: Acho que a concordância correta seria "caminhando de encontro"

garantia de uma maior qualidade dos serviços de TIC, e a sua gestão efetiva [MAGALHÃES 2007].

Já Governança de Tecnologia da Informação, Governança de TI ou Governança de TIC, é definida por alguns autores [ITGI 2009], [ISACA 2007], [ITSMF 2008] como um subconjunto da disciplina Governança Corporativa, centrado na tecnologia da informação (TI) e seus sistemas de desempenho e gestão de risco. O crescente interesse em governança TI é, em parte, devido a uma série de iniciativas que visam garantir a criação de mecanismos de auditoria e segurança confiáveis nas empresas, de modo a mitigar riscos aos negócios e evitar a ocorrência de fraudes (ou assegurar que haja meios de identificá-las), garantindo a transparência na gestão das empresas, como, por exemplo, Sarbanes-Oxley [REZZY 2007] nos EUA e Basileia II [BIS 2006] na Europa. Movimentos como estes demonstram como instituições de referência no mercado mundial reconhecem que os projetos de TIC podem facilmente sair de controle e afetar profundamente o desempenho de uma organização.

Comment [M909]: Acho que faltou um "de"...deveria ser "governança de TI"

Não desejamos aqui discutir em detalhes os êxitos ou melhorias que estas ferramentas têm alcançado (em especial ITIL e COBIT) para os processos de suporte ao core business de nossas organizações, contudo pretendemos explorar alguns contextos de aplicação destas.

Comment [M910]: Verificar se não seria adequado usar um termo em português ou então colocar o termo utilizado em itálico

23.2 Modelos de Gestão em TIC

Formatted: Bullets and Numbering

23.2.1 COBIT

Formatted: Bullets and Numbering

23.2.2ITIL

Formatted: Bullets and Numbering

23.2.3BSC

Formatted: Bullets and Numbering

O BSC (*Balanced Scorecard*) é um modelo de gestão, desenvolvido há 11 anos por Kaplan e Norton da Universidade de Harvard, para avaliar o desempenho estratégico e, consequentemente, gerir o sistema de estratégias de uma organização, sendo considerado uma das ferramentas de grande importância na área de planejamento estratégico com o objetivo de traduzir estratégia em ação.

Comment [M911]: Acho que seria interessante explicitar o ano em que foi desenvolvido o BSC, pois com o passar do tempo esse número vai mudar o que implicaria em uma informação errada para o leitor.

23.2.4IT Flex

Formatted: Bullets and Numbering

A metodologia IT Flex, com sua proposta de transformação da área de TI em uma provedora de serviços de forma continuada para a organização, parte da estruturação

do desempenho da área de TI que possibilita a ela a oportunidade de fornecer serviços de TI com toda a qualidade que os seus clientes requerem, com custos e níveis de serviço associados que alinhem TI às necessidades das diferentes áreas de negócio da organização [MAGALHÃES 2007].

Quando os serviços de TI estão alinhados aos objetivos estratégicos estabelecidos pela estratégia de negócio e otimizados para todo o ciclo de vida do serviço, a organização consegue associar os custos da área de TI ao valor produzido para o negócio, enxergando a verdadeira contribuição da área de TI. Isto é obtido, segundo Magalhães, através da aplicação da metodologia IT Flex conforme descrito a seguir:

Comment [M912]: Explicitar a fonte se [Magalhães 2007] ou outra.

- Menores custos e maior eficiência, integrando o Service Desk à toda a infra-estrutura de TI e gerenciando proativamente o portfolio de serviços de TI;

Comment [M913]: Colocar em itálico ou traduzir termo ao português.

23.2.5COSO

Formatted: Bullets and Numbering

Esta publicação tornou-se referência mundial para o estudo e aplicação dos controles internos. Posteriormente a Comissão transformou-se em Comitê, que passou a ser conhecido como C.O.S.O. - The Comitee of Sponsoring Organizations (Comitê das Organizações Patrocinadoras). O C.O.S.O. é uma entidade sem fins lucrativos e dedicada à melhoria dos relatórios financeiros através da ética, efetividade dos controles internos e governança corporativa [COCURULLO 2004].

Em 2002, o ato de Sarbanes-Oxley foi criado para restaurar confiança de investidores dos mercados públicos dos Estados Unidos, devastados por escândalos e lapsos nos negócio envolvendo governanças corporativas. Embora reescrevessem literalmente as regras de contabilização corporativa, bem como a sua divulgação, as páginas inumeráveis do ato da sustentação legal seguem uma premissa simples: a governança corporativa e as práticas éticas de negócio já não são mais opcionais em TI, mas são leis.

O ato Sarbanes-Oxley representa a parte a mais significativa de uma legislação sobre os negócios, desde a última metade do século, pois evidencia a contabilização corporativa. Entretanto, é importante enfatizar que a seção 404 não requer apenas que as empresas estabeleçam e mantenham uma estrutura interna adequada ao controle, mas avaliar também sua eficácia anualmente, ou seja, para aquelas organizações que começaram o processo de conformidade e que a TI exerce um papel vital suportando os componentes de sistemas, de dados e de infraestrutura e que são críticos no processo de relatório financeiro [COSO 2006].

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 é muito genérico, com visão de auditoria, muitas organizações usam o COBIT (Control Objectives foi 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 COBT, como guarda-chuva para diversas metodologias e melhores práticas indicadas para tecnologia da informação.

23.2.6ISO/IEC 20000

A ISO/IEC 20000 é a primeira norma editada pela ISO (International Organization for Standardization) que versa sobre gerenciamento de serviços de TI (Tecnologia da

Comment [M914]: Colocar em itálico.

Comment [M915]: Acho que faltou um "a" par a ficar assim "restaurar a confiança"

Comment [M916]: "nos negócios" é a concordância correta.

Comment [M917]: Acho que a concordância correta é "governança corporativa"

Comment [M918]: Acho que a concordância certa seria "mas avaliem também sua eficácia..."

Comment [M919]: Essa frase ficou muito longa, talvez fosse melhor dividir em duas pra facilitar o entedimento.

Comment [M920]: Colocar em itálico

Comment [M921]: Acho que a frase tem um "é" a mais...imagino que o certo é "é um modelo de trabalho muito genérico."

Comment [M922]: Acho que é "for" ao invés de "foi"

Comment [M923]: Colocar em itálico

Comment [M924]: Faltou um i em "COBIT"

Formatted: Bullets and Numbering

Comment [M925]: Colocar em itálico

A ISO 20000 é um conjunto que define as melhores práticas de gerenciamento de serviços de TI. O seu desenvolvimento foi baseado na BS 15000 (British Standard) e tem a intenção de ser completamente compatível com o ITIL (Information Technology Infrastructure Library). A sua primeira edição ocorreu em dezembro de 2005.

Comment [M926]: Colocar em itálico

Comment [M927]: Colocar em itálico

Foi desenvolvido para responder às necessidades de uma audiência global e fornecer um entendimento comum da gestão de serviços de tecnologias de informação em todo o mundo. Cobre os aspectos responsáveis por 80% do investimento total em tecnologias de informação da grande maioria das organizações. É publicado em duas partes e permite aos prestadores de serviços compreender como podem alcançar a qualidade no serviço prestado aos seus clientes, internos e externos. A certificação é o resultado da monitoração do nível de serviço face ao padrão definido, acrescentando valor real para as organizações não só porque demonstram a qualidade dos serviços internos como lhes permite selecionar parceiros externos adequados [ISO20000 2005].

Comment [M928]: Acho que a concordância correta seria "compreenderem"

23.2.7VAL IT

Formatted: Bullets and Numbering

- Maximizar a qualidade dos processos de negócios para TI permitir investimentos em negócios com particular ênfase para a definição dos principais indicadores financeiros, a quantificação dos "suaves" prestações e à avaliação global do risco de queda.

Comment [M929]: Acho que a concordância certa deveria ser "permitindo"

Comment [M930]: Não entendi essa parte...acho que ta faltando alguma coisa, ou a palavra "prestaçãoes" está sobrando

23.2.8CMMi sob a perspectiva de Governança de TI

Formatted: Bullets and Numbering

23.3oITIL

Formatted: Bullets and Numbering

23.3.1Definição

Formatted: Bullets and Numbering

A sigla ITIL significa Information Technology Infrastructure Library (ITIL, em português, quer dizer Biblioteca de Infraestruturas de Tecnologias da Informação). ITIL é uma compilação das melhores práticas e processos no planejamento, aprovisionamento e suporte de serviços de Tecnologia de Informação (TI) [ITIL 2009] e pode ser considerada um conjunto de boas práticas de governança organizado de forma sistemática, e portanto um framework. À medida que as empresas reconheceram a sua dependência crescente nas TI para conseguirem satisfazer os objetivos do negócio e irem de encontro às necessidades da empresa, muitos determinaram que a maior qualidade dos serviços de TI, e a sua gestão efetiva, era necessária [EUROCOM 2006].

Comment [M932]: Itálico

Comment [M933]: Itálico

Comment [M934]: Acho que a concordância fica melhor assim "dependência crescente da TI"

Existe uma grande divergência entre os autores sobre o uso do gênero do ITIL, se é "a" ITIL ou se é "o" ITIL. A terminologia "a" ITIL é utilizada quando o autor prefere se referir à Biblioteca de Infraestrutura de TI (tradução da sigla ITIL). Quando o autor se refere ao ITIL como framework, a denominação mais aplicada é "o ITIL". Neste capítulo estaremos nos referindo ao segundo caso.

Comment [M935]: itálico

23.3.2Histórico

Formatted: Bullets and Numbering

O Office of Government Commerce (OGC) originou a Versão 1 do ITIL, que foi chamada a GITIM, Government Information Technology Infrastructure Management. Esta Versão 1 é bastante diferente da versão atual. Parte desta diferença é devida à gradual maturidade do ITIL e às mudanças na indústria de TI. Entre o desenvolvimento da Versão 1 e o ano 2001, o número de documentos (livros) utilizados no ITIL cresceu para mais de 32. No ano 2000, a Microsoft utilizou ITIL como a base para o

Comment [M936]: itálico

Comment [M937]: Acho que faltou incluir um "o" pra ficar assim "utilizou o ITIL"

(MOF). No ano 2000 também se pôde presenciar a CCTA passar a ser o OGC [ITIL 2009].

Com a versão atual do ITIL, Versão 3 lançada em 2007, uma das principais deficiências corrigidas foi um incremento em matérias que ajudem a identificar o retorno dos investimentos em TI. Um problema muito freqüente em governança de TI que era normalmente indicado como um problema para a adoção efetiva do ITIL. Embora tenha sido atualizada com as necessidades atuais, a nova versão é ainda mais concisa do que a versão anterior, reduzida para 5 livros principais que compõem seu núcleo e vários outros livros que poderão complementar o ITIL posteriormente.

Comment [M939]: Acho que a concordância correta seria "ajudam"

Comment [M940]: A utilização do trema foi descartada de acordo com a nova ortografia brasileira

Comment [M941]: Acho que seria escrito melhor assim "atualizada com as necessidades correntes" pra evitar a repetição da mesma palavra.

Formatted: Bullets and Numbering

23.3.3Regulamentação do ITIL

- Information Technology System Management Forum (itSMF)

Comment [M942]: Itálico

Formatted: Bullets and Numbering

23.3.3.1•Certificações / Treinamentos

Dentro do ITIL existe um número de certificações individuais. O dono da certificação e dos testes da certificação é a Fundação Holandesa Exameninstituut voor Informatica - EXIN [EXIN 2009] e o grupo Britânico Information Systems Examination Board - ISEB [ISEB 2009]. O EXIN e o ISEB desenvolveram juntos o sistema profissional de certificação para o ITIL. Isto foi feito em cooperação com o OGC [OGC 2009] e o Information Technology System Management Forum [ITSMF 2009].

Comment [M943]: Achei que o termo dono não ficou legal...acho que ficaria melhor assim "A responsável pela concessão da certificação e da aplicação dos testes de certificação"

Comment [M944]: itálico

Comment [M945]: itálico

Existem três certificações reconhecidas individualmente: i) Foundation Certificate in IT Service Management; ii) Practitioner Certificate in IT Service

Em adição às certificações individuais existe para as organizações uma certificação, BS15000, que é o primeiro padrão mundial para a IT service management (Gestão de Serviços de TI). Este padrão especifica um conjunto de processos de gestão interrelacionados e é baseado no ITIL.

Comment [M947]: itálico

23.3.3.2 **Direitos Autorais**

Formatted: Bullets and Numbering

O Office of Government Commence, cujo web site pode ser visto em [OGC 2009], é o "dono" 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 das TI, estendendo estas a outras áreas diversas.

Comment [M948]: "Commerce" e não "Commence"

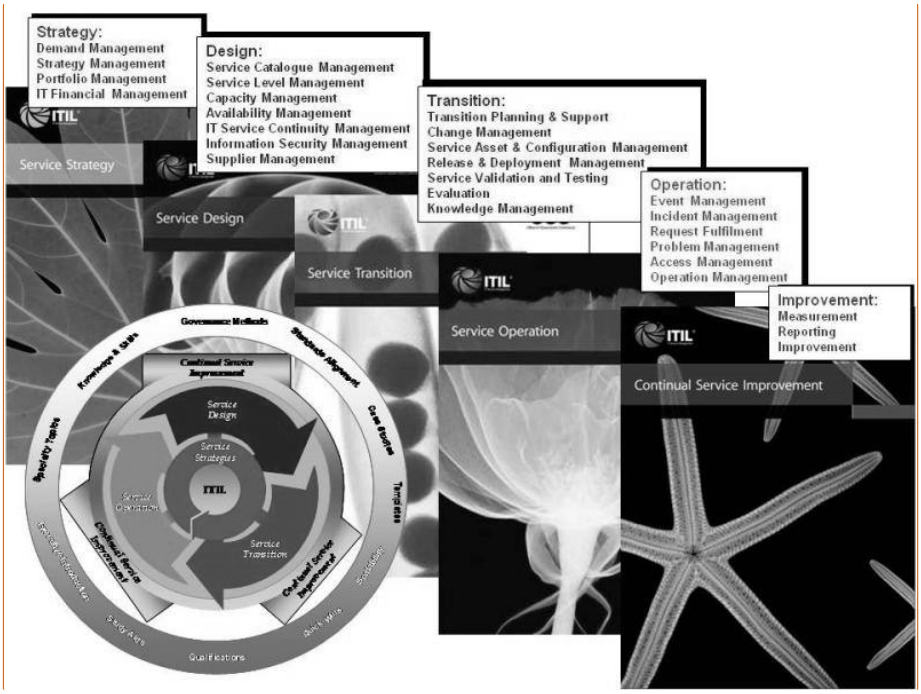
Comment [M949]: itálico

Comment [M950]: Acho que ficaria melhor colocar "detentor dos direitos do ITIL" ao invés de "dono"

Comment [M951]: "Melhoria da TI"

23.3.3.3 **Publicação de Conteúdos Oficiais**

Formatted: Bullets and Numbering



Comment [M952]: A figura deve estar em português e a qualidade da imagem não está legível o suficiente para os leitores

Formatted: Bullets and Numbering

23.3.3.4 **Fórum de Fomento (iTSMF)**

23.3.4Estrutura do ITIL

Formatted: Bullets and Numbering

Sendo considerado o centro dos livros que compõem o núcleo o ITIL versão 3, esse livro alinha a tecnologia ao negócio, transformando as estratégias de negócio em estratégias de TI. Seus principais objetivos são a definição de papéis e responsabilidades, definição das estratégias de serviços, ligação dos planos de negócios à planos de TI, planejamento de custos e riscos de investimentos em TI.

Comment [M953]: Acho que seria "núcleo do ITIL"

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 fator de sucesso.

Comment [M954]: Acho que ficaria melhor "implementação e fatores de sucesso"

Esse volume trata do processo envolvido em administrar um aprimoramento contínuo dos serviços, assim como também administrar a interrupção dos serviços. O objetivo básico mostrar como aprimorar o serviço que já está implementado. O livro cobre tópicos como os princípios do CSI (*Continual Service Improvement*), papéis e responsabilidades, componentes necessários, os benefícios, a implementação, métodos, práticas e ferramentas assim como outras práticas relacionadas à CSI.

Comment [M955]: Acho que faltou um "é" pra ficar assim..."O objetivo básico é mostrar..."

A principal vantagem da aproximação do ITIL às "melhores práticas" é que os processos descritos são genéricos – aplicam-se independentemente da tecnologia

Comment [M956]: Acho que seria melhor assim

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 prativa, satisfazendo assim clientes e usuários [ITSMF 2008].

A Figura a seguir mostra o escopo do framework ITIL, que possui sete domínios e a representação do gerenciamento de serviços como ponto central do framework. O framework ITIL tem o propósito de fornecer uma integração entre a TI e os objetivos de negócio da organização, através de um gerenciamento da estrutura e do fornecimento e suporte dos serviços de TI [ITSMF 2008].

Comment [M957]: Não sei qual seria a palavra que se encaixa aqui se "prática" ou "proativa"

Comment [M958]: itálico

Comment [M959]: itálico

Comment [M960]: itálico

Comment [M961]: Figura deve estar em português

23.3.5 O que não é ITIL

A atenção crescente que o ITIL tem recebido é visto pela EMA (Enterprise ManagementAssociates) como um incentivo às organizações no sentido de conseguirem fornecer os serviços de TIC, tão essenciais ao negócio, cada vez de forma mais eficaz [DROGSETH 2004].

Contudo o crescimento do interesse precisa ser acompanhado com cautela, evitando que o ITIL seja visto como uma "solução mágica" para todos os problemas de TIC das organizações. Embora o ITIL tenha se mantido "aberto" à evolução de suas práticas, é necessário estabelecer um paralelo entre a proposição de um framework de referência e a dinâmica de sua aplicação nas organizações. Esta dicotomia, muitas vezes não é bem percebida pelo mercado a ponto de gerar certa confusão às organizações no processo de seleção de serviços de consultoria e produtos de software que afirmam ser "*ITIL-compliant*", mas cuja orientação dos fornecedores sobre o que este termo representa e significa, muitas vezes não é claro o suficiente para os compradores [DROGSETH 2004].

- ITIL não é uma metodologia para implementar processos de Gestão de Serviços de TI – é uma framework flexível que permite adaptar-se para ir ao encontro das necessidades especificas;

Formatted: Bullets and Numbering

Comment [M962]: itálico

Comment [M963]: italico

Comment [M964]: "um framework"

Comment [M965]: itálico

23.3.6Fronteiras com outros modelos e limitações

Formatted: Bullets and Numbering

- MOF – É o Microsoft Operations Framework, baseado na versão 2 do ITIL.
- HP ITSM – É o Hewlett & Packard IT Service Management Reference Model um modelo proprietário da HP também baseado na versão 2 do ITIL.
- 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].

Comment [M966]: itálico

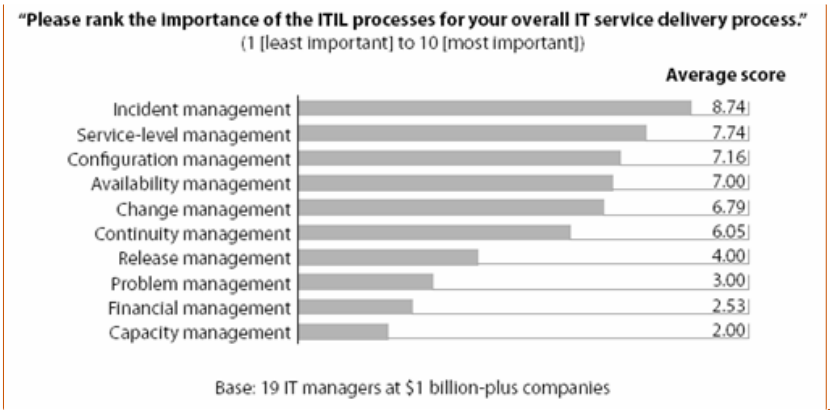
Comment [M967]: itálico

Comment [M968]: itálico

- **ITIL não atendia à visão da organização na era .COM -** Esta limitação foi característica de versões anteriores, e foi relativamente resolvida com o advento da versão 3. Em sua versão original o ITIL não levava em consideração a empresa estendida ou o fato de que muitas organizações que prestavam serviços internos de TI, atualmente, tem que integrar múltiplos parceiros de serviços externos em seus sistemas de

23.3.7Ponto de Partida

Formatted: Bullets and Numbering



Comment [M970]: Figura deve estar em português

Figura 23.4 - Rank de processos de ITIL de acordo com a importância.
Fonte: [MENDEL & PARKER 2005]

Comment [M971]: Acredito que a palavra seja "Ranking"

Podemos observar que no topo do ranking aparece a Gerenciamento de Incidentes, que lida com a resolução imediata da indisponibilidade de serviços. Esta colocação não é surpreendente, considerando que faz parte do contexto imediato de construção de um processo estruturado para reação a crises, gerenciar incidentes. Por isso faz sentido ser o ponto de partida [DUBIE 2005].

Comment [M972]: "aparece o gerenciamento"

A Gerência de Disponibilidade e a Gerência de Mudança foram os seguintes colocados. Estes processos são os pré-requisitos para um Gerenciamento de Incidentes e Gerenciamento do Nível de Serviço "bem sintonizados" [DUBIE 2005].

Comment [M973]: Acho eu esse termo não ficou bom "seguintes colocados" se possível pensem em algo diferente.

Formatted: Bullets and Numbering

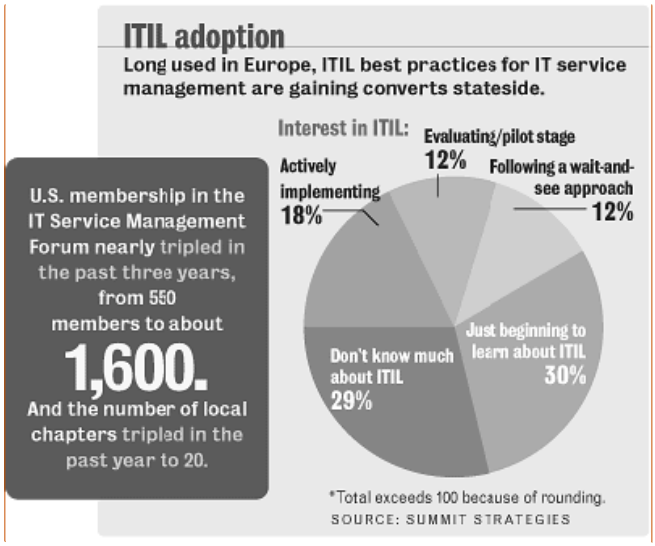
23.3.8Comentários sobre práticas de sucesso

- **Priorizar a implantação de Gestão de Configurações:** a Gestão de Configurações fornece uma base para mapear os componentes de infraestrutura de TIC, adotar o uso de ferramentas de identificação e

deste processo é possível, inclusive identificar o nível de dependência entre os serviços e seus insumos.

23.3.9Público Alvo

Formatted: Bullets and Numbering



Comment [M974]: Figura deve estar em português

- United Health Group em Minneapolis

Comment [M975]: itálico

Formatted: Bullets and Numbering

23.3.10Utilização do ITIL

23.4 COBIT

O COBIT - ***C**ontrol **Ob**jectives for **I**nformation and related **T**echnology* representa a visão de um grupo de experts focado no estudo da Governança em TIC. Trata-se de um conjunto de boas práticas sobre processos de gerenciamento da TI nas organizações, que aborda desde aspectos técnicos, até processos e pessoas. Sua estrutura é organizada em processos que são interligados com o objetivo o controlar a TI, através da formação de um framework de controle que tem o propósito de assegurar que os recursos de TI estarão alinhados com os objetivos da organização.

O princípio do framework do COBIT é vincular as expectativas dos gestores de negócio com as responsabilidades dos gestores de TI. Assim, busca fazer com que a TI seja mais suscetível ao negócio. O COBIT não se trata de um padrão definitivo, ele tem que ser adaptado para cada organização.

Formatted: Bullets and Numbering

Comment [mdp976]: itálico

Comment [mdp977]: itálico

Comment [mdp978]: itálico

- Ajuda identificar quais recusros de TI devem ser alavancados com maior efetividade;

Comment [mdp979]: "recursos" trocar a ordem das letras "s" e "r"

23.4.2Histórico

Formatted: Bullets and Numbering

O COBIT foi criado para atender a necessidade de um framework de controle de TI compreensivo para o negócio, gerência de TI, auditores, e eliminar as disparidades de controles e guias de avaliação.

Comment [mdp980]: itálico

		ITGI - IT Governance Institute – incluir normas e guias associadas à gestão. O ITGI passa a ser o principal editor da framework

Comment [mdp982]: Acho que é "inclui" sem o r no final

23.4.3Regulamentação do COBIT

Formatted: Bullets and Numbering

23.4.3.1 **Certificações / Treinamentos**

Formatted: Bullets and Numbering

23.4.3.2 **Direitos Autorais**

Formatted: Bullets and Numbering

23.4.3.3 **Publicação de Conteúdos Oficiais**

Formatted: Bullets and Numbering

- Framework - Identifica como os critérios da informação e os recursos de TI são importantes para suportar os objetivos de negócios.

Comment [mdp983]: itálico

23.4.3.4 **Fórum de Fomento (ISACA)**

Formatted: Bullets and Numbering

23.4.4Estrutura do COBIT

Formatted: Bullets and Numbering

23.4.4.1 **Primeira Dimensão do Cubo – Processos de TI**

Formatted: Bullets and Numbering

23.4.4.2 **Segunda Dimensão do Cubo – Critérios de Informação**

Formatted: Bullets and Numbering

- Disponibilidade: A informação deve ser disponível quando requerida pelo processo de negócio agora e no futuro, e deste mode deve ser salva, guardada enquanto recurso.

Comment [mdp984]: acho que o certo é "modo"

- Confiabilidade: A informação deve ser provida de forma apropriada,

financeiros para seus usuários e órgãos fiscalizadores, conforme leis e regulamentos.

23.4.4.3 Terceira Dimensão do Cubo – Recursos de TI

Formatted: Bullets and Numbering

23.4.5Não é COBIT

O COBIT é um framework de controle com Diretrizes de Auditoria, Então ele [ITGI 2007]:

> O COBIT trata-se de um conjunto de diretrizes baseadas em auditoria para processos, práticas e controles de TI, voltado para redução de risco, enfoca integridade, confiabilidade e segurança

Formatted: Bullets and Numbering

Comment [mdp985]: itálico

Comment [mdp986]: acho que ficaria melhor "focando em" ao invés de "enfoca"

23.4.6Fronteiras com outros modelos

O COBIT é um *framework* único, não tendo outro similar, pois ele acomoda os padrões internacionais mais importantes e é reconhecido como um padrão de fato para o controle de TI. Muitas empresas acham conveniente usar o COBIT por ele se relacionar com outros frameworks, tais como COSO, ITIL, ISO 17799, CMM e PMBOK [TIEXAMES 2009]. Ele diz o que tem de ser feito, e não se preocupa em como fazer. Cobre todos os processos do ITIL, entretanto o ITIL é mais detalhado. Atende os requisitos regulatórios nos quais a empresa está submetida, por isto pode ser utilizado para cumprir a conformidade com a Sarbanes Oxley.

O COBIT está em um nível mais genérico, por isto pode ser utilizado para avaliar outros processos implementados por outros frameworks como a ISO 17799. Ele também pode ser aplicado depois que outros padrões a nível mais operacional já estejam aplicados, servindo para auditar estes processos. Encontra-se alinhado com o COSO, que é um framework para controle de interno, não somente de TI, pode ser utilizado em qualquer área de negócio. A Tabela 23.3 mostra a relação do COBIT com os outros frameworks:

	serviços de TI. Ele é focado em "como" deve ser os serviços e os processos de TI.
	O SEI(Software Engineering Institute) é a organização que desenhou o Capability Maturity Model (CMM). Este modelo ajuda as empresas a melhorem seus processos de entrega de software e controle de processos.
	O Framework COSO é uma padrão aceito para estabelecer controles internos na empresa e determinar sua eficácia, pode ser aplicado a TI como também a qualquer área da empresa.

Formatted: Bullets and Numbering

Comment [mdp987]: itálico

Comment [mdp988]: itálico

Comment [mdp989]: itálico

Comment [mdp990]: Acho que a concordância correta seria "devem"

Comment [mdp991]: itálico

Comment [mdp992]: itálico

Comment [mdp993]: Acho que a palavra correta aqui seria "melhorarem" e não "melhorem"

Comment [mdp994]: itálico

23.4.7Ponto de Partida

Um ponto de partida para que as organizações interessadas em aplicar o modelo de governança do COBIT pode ser encontrado no COBIT *Management Guidelines*, que provê uma ferramenta distinta para cada um dos 34 processos do COBIT, que é o modelo de maturidade, semelhante ao CMMI, com níveis de 0 (Não existente) a 5 (Otimizado) onde em cada nível existe uma descrição de como devem estar dispostos os processos para alcançá-los. Além disso, este modelo pode ser utilizado como um checklist para identificar melhorias nos processos de TI existentes na organização.

Geralmente, estes níveis de maturidade são utilizados para uma organização definir rapidamente, com base nos cenários descritos, em que nível se encontra e em que nível pretende chegar futuramente. Na maior parte das vezes, a aplicação deste modelo é feita através de reuniões com os gestores, onde pede-se que este identifiquem o nível atual e o desejado dos processos [ALVES & RANZI 2006].

A Figura 23.12 ilustra os níveis do modelo de maturidade utilizado pelo COBIT, enquanto que a Tabela 23.4 mostra significado de cada nível do modelo genérico do COBIT.

Formatted: Bullets and Numbering

Comment [mdp995]: itálico

Comment [mdp996]: Acho que a concordância certa é "estes"

Comment [mdp997]: Acho que faltou um "o" pra ficar assim "mostra o significado"

	Já existem processos, só que não documentados; não existe padrões.
	Processos em aperfeiçoamentos, já fornecem as boas práticas. Mas falta ferramentas de automação.

Comment [mdp998]: Acho que a concorrência certa é "existem"

Comment [mdp999]: "Acho que o certo é "aperfeiçoamento"

Comment [mdp1000]: Acho que a concordância correta é "faltam"

23.4.8Comentários sobre práticas de sucesso

Formatted: Bullets and Numbering

23.4.9Público Alvo

Formatted: Bullets and Numbering

cadeia produtiva da empresa. Ele foi projetado para ser utilizado por basicamente três públicos distintos [ITGI 2007]:

23.4.10 Utilização do COBIT

Formatted: Bullets and Numbering

O COBIT ainda não está sendo amplamente utilizado nas organizações mundialmente. Pode-se comprovar esta informação através do resultado de uma pesquisa realizada pela *International Network Services* com 194 organizações de todo o mundo, apresentado na Figura 23.6 da seção 23.3.10. Essa pesquisa mostrou que apenas 7% das organizações mundiais responderam que utilizam a COBIT.

Comment [mdp1001]: Acho que é "o" ao invés de "a"

Entretanto, uma pesquisa realizada recentemente, em 2009, pela FGV-SP confirma a aderência das organizações de TI ao COBIT. Segundo a pesquisa, o COBIT é a principal prática de governança de TI utilizada no Brasil. O gráfico da Figura 23.13 ilustra os resultados dessa pesquisa [FGV 2009].Dentre as empresas brasileiras que adotaram este modelo de governança de TI, podemos citar: a Empresa Brasileira de Correios e Telégrafos (ECT) [COMPUTAÇÃO CORPORATIVA 2009], a empresa de Call Center Contax [INFO 2008], a secretaria de Ti do Supremo Tribunal Federal (STF) [CONIP 2008], a Controladoria Geral da União (CGU) [RODRIGUES 2009], o conglomerado petroquímico Braskem [INFO 2009], a Petrobrás, a Trevisan & Associados, a GOL [REVISTA FATOR BRASIL 2008] e o Banco Central [LINHA DE CÓDIGO 2007].

Comment [mdp1002]: "TI" em maiúsculo

23.5 Iniciativas de Integração dos Principais Modelos

Em uma implantação de Governança de TI de uma organização, é possível a utilização de frameworks que abordam aspectos semelhantes, relacionados com as áreas foco da governança, tais como: Alinhamento Estratégico, Entrega de Valor, Gerência de Recursos, Gerência de Riscos e Medição de Desempenho.

Formatted: Bullets and Numbering

Comment [mdp1003]: itálico

	PO9, DS5, DS11, DS12, MA2,

Comment [mdp1004]: Acho que a tabela deveria ficar centralizada na página

- O COBIT ajuda a vincular as melhores práticas do ITIL aos os requisitos de negócio e aos responsáveis do processo de TI;

Comment [mdp1005]: Acho que esse "os" está sobrando na frase e deveria ser excluído

Formatted: Bullets and Numbering

23.6 Implantação de Modelos de Gestão

Atualmente o ambiente do negócio exige que as unidades de TI estejam mais atentas às

melhorar esta vertente cada vez mais empresas adotam a hipótese de implantação de governança de TIC.

Com base num processo de análise qualitativa e comparativa dos mencionados estudos de caso [TECHREPUBLIC 2002], [TECHREPUBLIC, 2003], conseguimos identificar os passos seguidos em cada um dos dois estudos de caso, e concluir que, mesmo com algumas peculiaridades, o sequenciamento da ações respeitaram de forma consistente, em cada caso, um conjunto comum de passos. Uma síntese do resultado desta avaliação pode ser ilustrado na Figura 23.13.

Comment [mdp1006]: Acho que a concordância certa é "das"

Comment [mdp1007]: A concordância certa é "ilustrada" já que o termo refere-se a síntese

Após uma análise do processo destes dois estudos de caso, verificamos que não existem fórmulas mágicas nem técnicas especiais de implantação. Os resultados da avaliação da maturidade dos processos fornecem um ponto de partida. Mas, a compreensão das relações entre esses processos, serviços e os objetivos traçados pela a organização vão ajudar a determinar a ordem "correta" da implantação das melhores práticas de governança.

Comment [mdp1008]: Esse "a" ta sobrando na frase e deve ser excluído

23.7 Tópicos de Pesquisa

Formatted: Bullets and Numbering

- [GREFEN-MEHANDJIEV&KOUVAS 2009]. P Grefen, N Mehandjiev, G Kouvas, G. Dynamic Business Network Process Management in Instant

em: http://www.exodus.gr/Documents/BETA%20WP198.pdf Acesso em: 30/09/2009.

23.8 Sugestões de Leitura

Formatted: Bullets and Numbering

Alexandre J. H. de O. Governança Ágil em TIC: Um Modelo para apoio à Governança de Tecnologia da Informação e Comunicação inspirado no Paradigma das Metodologias Ágeis. Programa de Pós-graduação *stricto sensu* em Ciência da Computação. Centro de Informática, Universidade Federal de Pernambuco. Dissertação de Mestrado. Disponível em: <www.cin.ufpe.br/~ajhol/publicacoes>. Acesso em: 17/12/2009.

- Se houver interesse em aprofundar o tema Governança Ágil em TIC, sugerimos a leitura do artigo:[LUNA 2009b]. de OLIVEIRA LUNA, Alexandre J. H.; COSTA, Cleyverson P.; de MOURA, Hermano P.; NOVAES, Magdala A.; do NASCIMENTO, César A. D. C. ; Governança Ágil de TIC: rompendo paradigmas. JISTEM - Journal of Information Systems and Technology Management; 2009. Disponível em: < http://www.jistem.fea.usp.br/index.php/jistem/issue/archive >. Acesso em: 17/12/2009.

- Se tiver algum interesse em maiores informações sobre comparação entre os dois principais modelos de governança de TIC – ITIL e COBIT, é interessante ler o artigo CLEMENTI,S.; CARVALHO, T. Governança de TI: Comparativo entre COBIT e ITIL. Anais do Congresso Anual de Tecnologia da Informação – CATI. São Paulo, 2004.

- Para ser set uma visão geral do framework ITIL, um bom começo é a leitura do artigo: [ITSMF 2008]. itSMF - IT Service Management Forum. An Introductory Overview of ITIL® V3. Disponível em: <http://www.best-management-practice.com/gempdf/itSMF_An_Introductory_Overview_of_ITIL_V3.pdf>. itSMF , 2008. Acesso em: 23/01/2009.
- Pode-se obter a versão completa do COBIT 4.1, em diversos idiomas através da referência: [ISACA 2007]. COBIT - Control Objectives for Information and

Comment [mdp1009]: Corrigir esta data de acesso

Comment [mdp1010]: Corrigir a data de acesso

Comment [mdp1011]: Verificar se isto está correto?

Comment [mdp1012]: Acho que o correto seria "Para se ter"

<http://www.isaca.org/Content/NavigationMenu/Members_and_Leaders/COBIT6/Obtain_COBIT/Obtain_COBIT.htm>. Acesso em: 13/01/2009.

23.9 Exercícios

Formatted: Bullets and Numbering

9. ( ) São princípios da governança de TI: responsabilidade corporativa, prestação de contas equidade e transparência.

Comment [mdp1013]: Acho que faltou um "com" pra ficar assim "prestação de contas com equidade e transparência"

3. ( ) Os 4 domínios possuem 34 processos, estes processos especificam o que o negócio precisa para alcançar seus objetivos. A entrega de informação é controlada por 68 objetivos de controle de alto nível, dois para cada processo.

Comment [mdp1014]: Faltou indicar de que modelo são estes domínios...ou coloca no título do questionário que esse V ou F é referente ao COBIT ou especifica nas afirmações como um todo.

2. ( ) Os graus de controle são indicados em cada processo para cada aspecto de critérios de informação e recursos de TI.

Comment [mdp1015]: Mais uma vez faltou indicar a que modelo se refere esses graus de controle...acho que o mais adequado é citar antes de começar os exercícios que as 5 primeiras questões se referem ao ITIL e as 5 últimas ao COBIT

23.10 **Referências**

Formatted: Bullets and Numbering

BALL, E. S., “ITIL: What it is and why you should care”, 2005. Disponível em: <http://images.globalknowledge.com/wwwimages/whitepaperpdf/WP_ITIL.pdf >.Acesso em: 01/10/2009.

BERG, C., Value-DrivenIT,valuedrivenit.com. Cliff Berg Imprints, Reston VA, USA, 2008. Disponível em: <http://valuedrivenit.com/downloads/Value-Driven_IT.pdf>. Acesso em: 30/09/2009.

COMPUTERWORLD ONLINE. Construtora centraliza gestão dos chamados e melhora suporte. 20/08/2009. Disponível em: <

Comment [mdp1016]: Verificar se o modelo de referência a ser utilizado está correto

http://computerworld.uol.com.br/gestao/2009/08/19/construtora-centraliza-gestao-dos-chamados-e-melhora-suporte>. Acesso em: 02/10/2009.

Page 104: [1] Comment [A795]	Alinne	10/24/2009 1:42:00 PM

Acho que pode juntar com o paragrafo anteriror 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.

Page 104: [2] Comment [A796]	Alinne	10/24/2009 1:43:00 PM

Acho que ficaria melhor assim:

De acordo com os conceitos expostos, ficou clara a definição do termo métrica.

Page 104: [3] Comment [A797]	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.