



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA



CURSO CIÊNCIA DA COMPUTAÇÃO  
TURMA 2004.2

---

# Modelagem e Avaliação do CMMI no SPEM para Definição de um Meta-Processo de Software

**Autor**

*Rodrigo Cavalcante Mendes (rcm2@cin.ufpe.br)*

**Orientador**

*Prof.º Alexandre Marcos Lins de Vasconcelos*

**Recife, Março 2005**

## AGRADECIMENTOS

Este trabalho reflete a conclusão de uma fase importante e marcante em minha vida, pois representa o resultado de um grande esforço de pesquisa, aplicação e doação, numa área de meu interesse na qual estarei sempre tentando adquirir novos conhecimentos e promovendo sua evolução. Marca sim o término de mais uma fase – graduação – e início de outra fase em minha vida acadêmica e profissional.

Agradeço ao meu Pai, Jorge Mendes Lima e minha Mãe, Maria do Socorro Cavalcante Mendes, pelo suporte, compreensão e dados durante minha jornada, e meus irmãos Rafael Cavalcante Mendes e Roberta Vaz de Araújo.

Agradeço de maneira especial ao Prof. Alexandre Vasconcelos e a Sandro Oliveira pela disponibilidade, receptividade e orientações constantes, que tornaram realização deste trabalho possível.

Agradeço a todos que constituem o Centro de Informática, que possibilitaram minha formação em Ciências da Computação, num dos melhores centros de referência do país.

Um agradecimento especial para alguns amigos que me incentivaram e participaram tanto em minha vida acadêmica como pessoal, obrigado, Ana Carolina do Amaral, Bárbara Siqueira, Cibele Christ, Eduardo Salvador, Felipe Pereira, Heitor Vital, Jennifer Lima, Marcílio Gomes Filho, Rangner Ferraz e Vitor Wanderley e a todos que direta ou indiretamente me apoiaram durante toda minha jornada.

# **MODELAGEM E AVALIAÇÃO DO CMMI NO SPEM PARA DEFINIÇÃO DE UM META-PROCESSO DE SOFTWARE**

Este trabalho de graduação propõe um estudo detalhado das normas e modelos de qualidades – CMMI, SPEM – visando modelar o CMMI no SPEM, para capturar informações sobre a aderência e compatibilidade, capturando os componentes do processo de software e os seus relacionamentos baseando-se no mapeamento deste modelo.

A partir desse estudo, é possível definir, agregando contribuições de normas como ISO 9000-3, um Meta-Processo para determinar uma terminologia única para a definição de processos de software em um Ambiente, deve ser capaz de permitir a manutenção a fim de atender as necessidades de futuras modificações (novas versões) nestes modelos/normas ou a inclusão de características de outros modelos/normas.

# ÍNDICE

AGRADECIMENTOS .....	2
1 INTRODUÇÃO .....	12
1.1 Contexto.....	12
1.2 Objetivos .....	13
1.3 Estrutura do Trabalho .....	16
2 O MODELO CMMI.....	18
2.1 Capability Maturity Models (CMMs).....	18
2.2 CMMI – CMM Integration .....	20
2.3 Disciplinas (áreas de Conhecimento) do CMMI .....	21
2.3.1 Engenharia de Sistemas .....	21
2.3.2 Engenharia de Software .....	21
2.3.3 Produtos Integrados e Desenvolvimento de Processos..	21
2.3.4 Fornecimento de Recursos.....	22
2.4 Tipos de Representações.....	22
2.4.1 Representação Contínua.....	22
2.4.2 Representação por Estágios .....	23
2.5 Categorias dos Componentes de uma Área de Processo....	23
2.5.1 Componentes Requeridos.....	23
2.5.2 Componentes Esperados .....	24
2.5.3 Componentes Informativos.....	24
2.6 Componentes Relacionados.....	24
2.6.1 Process Area .....	25

2.6.2	Purpose Statement .....	25
2.6.3	Introductory Notes .....	25
2.6.4	Related Process Areas.....	26
2.6.5	Specific Goals.....	26
2.6.6	Generic Goals.....	26
2.6.7	Specific Practices .....	27
2.6.8	Generic Practices .....	27
2.6.9	Typical Work Products .....	27
2.6.10	Subpractices.....	28
2.6.11	Generic Practice Elaborations .....	28
2.7	Considerações Finais.....	28
3	O MODELO SPEM .....	29
3.1	Arquitetura de Modelagem OMG.....	29
3.2	O pacote SPEM Foundation .....	30
3.2.1	Sub-pacotes do SPEM_Foundation .....	30
3.3	Modelo conceitual .....	31
3.4	Estrutura de Pacotes.....	32
3.4.1	Basic Elements (Elementos Básicos).....	33
3.4.2	Dependencies (Dependências) .....	35
3.4.3	Process Structure (Estrutura do Processo) .....	37
3.4.4	Process Components (Componentes de Processo).....	38
3.4.5	Process LifeCycle (Ciclo de vida do Processo) .....	40
3.5	Estereótipos do SPEM Profile.....	42
3.6	Considerações Finais.....	44
4	MAPEAMENTO CMMI E SPEM NO PROCESSO .....	45

4.1	Representações do Processo.....	45
4.1.1	Processo .....	45
4.1.2	Modelo de Ciclo de Vida .....	45
4.1.3	Combinação .....	45
4.1.4	Atividade.....	46
4.1.5	Artefato .....	46
4.1.6	Procedimento .....	46
4.1.7	Recurso.....	46
4.1.8	Padrão de atividades .....	46
4.1.9	Paradigma de Desenvolvimento .....	47
4.1.10	Tecnologia de Desenvolvimento .....	47
4.1.11	Restrições .....	47
4.2	Mapeamento Processo x SPEM x CMMI.....	47
4.3	Considerações Finais.....	51
5	ESTUDOS DE CASO.....	52
5.1	Gerenciamento de Requisitos .....	52
5.1.1	SG 1. Gerenciar Requisitos.....	53
5.1.2	SP 1.1–1. Obter um entendimento dos requisitos.....	53
5.1.3	SP 1.2–2. Obter o acordo dos requisitos .....	55
5.1.4	SP 1.3–1. Gerenciar mudanças nos requisitos .....	56
5.1.5	SP 1.4–2. Manter a rastreabilidade bidirecional dos requisitos	57
5.1.6	SP 1.5–1. Identificar inconsistências entre o plano de projeto e os requisitos .....	58
5.1.7	GG 2. Institucionalizar Processos Gerenciados .....	60

5.1.8	GP 2.1. Estabelecer uma política organizacional .....	60
5.1.9	GP 2.2. Planejar o Processo.....	61
5.1.10	GP 2.3. Fornecer Recursos .....	61
5.1.11	GP 2.4. Atribuir Responsabilidade.....	61
5.1.12	GP 2.5. Treinar Pessoas .....	61
5.1.13	GP 2.6. Gerenciar Configurações.....	62
5.1.14	GP 2.7. Identificar e envolver os skakeholders mais importantes	62
5.1.15	GP 2.8. Monitorar e controlar o processo .....	63
5.1.16	GP 2.9. Objetivamente Avaliar a Aderência.....	63
5.1.17	GP 2.10. Revisar o status com um maior nível de gerenciamento.....	63
5.2	Gerência de Configuração .....	64
5.2.1	SG 1. Estabelecer baselines.....	65
5.2.2	SP 1.1–1. Identificar os Itens de Configuração.....	65
5.2.3	SP 1.2–2. Estabelecer um sistema de gerência de configuração	66
5.2.4	SP 1.3–1. Criar ou liberar baselines.....	68
5.2.5	SG 2. Rastrear e controlar mudanças .....	70
5.2.6	SP 1.3–1. Criar ou liberar baselines.....	70
5.2.7	SP 2.2–1. Controle dos itens de configuração .....	71
5.2.8	SG 3. Estabeler integridade.....	73
5.2.9	SP 3.1–1. Estabelecer registros de gerência de configuração	73
5.2.10	SP 3.2–1. Realizar auditorias de configuração .....	74

5.2.11	GG 2. Institucionalizar Processos Gerenciados .....	76
5.2.12	GP 2.1. Estabelecer uma política organizacional .....	76
5.2.13	GP 2.2. Planejar o processo .....	76
5.2.14	GP 2.3. Fornecer recursos .....	77
5.2.15	GP 2.4. Atribuir responsabilidades .....	77
5.2.16	GP 2.5. Treinar pessoas .....	77
5.2.17	GP 2.6. Gerenciar configurações .....	78
5.2.18	GP 2.7. Identificar e envolver os stakeholders relevantes	78
5.2.19	GP 2.8. Monitorar e controlar o processo .....	78
5.2.20	GP 2.8. Monitorar e controlar o processo .....	79
5.2.21	GP 2.9. Avaliar objetivamente a aderência .....	79
5.2.22	GP 2.10. Revisar status em um maior nível de gerenciamento .....	80
5.3	Considerações Finais .....	80
6	TRABALHOS FUTUROS E CONCLUSÃO .....	81
6.1	Trabalhos Futuros .....	81
6.2	Conclusão .....	81
	REFERÊNCIAS .....	83



## ÍNDICE DE FIGURAS

Figura 1. Esquema utilizado pelo ambiente na adaptação do meta-modelo de processo.....	14
Figura 2. As três dimensões críticas. ....	18
Figura 3. Componentes do Modelo CMMI.....	25
Figura 4. Arquitetura de modelo definido pela OMG. ....	29
Figura 5. SPEM Foundation: subconjunto da UML 1.4. ....	30
Figura 6. Modelo Conceitual de uma classe. ....	32
Figura 7. Estrutura do Pacote SPEM.....	33
Figura 8. Pacote de Elementos Básicos.....	33
Figura 9. Dependências no SPEM. ....	35
Figura 10. Elementos do pacote Process Structure. ....	37
Figura 11. Pacote de Componentes de Processo. ....	39
Figura 12. Pacote de Ciclo de vida do Processo.....	41
Figura 13. Estrutura da P.A. de Gerência de Requisitos no SPEM.....	52
Figura 14. SP 1.1-1 representada no SPEM. ....	54
Figura 15. Representação da SP 1.2-2 no SPEM. ....	55
Figura 16. Representação no SPEM da SP 1.3-1. ....	56
Figura 17. SP 1.4-2 representada no SPEM. ....	57
Figura 18. SP 1.5-1 representada no SPEM. ....	59
Figura 19. P.A. de Gerência de Configuração nos estereótipos do SPEM...	64
Figura 20. SP 1.1-1 representada no SPEM. ....	65
Figura 21. A Prática Específica SP 1.2-2 no SPEM. ....	67

Figura 22. SP 1.3-1 representada no SPEM. ....	68
Figura 23. A SP 2.1-1 representada no SPEM. ....	70
Figura 24. Representação da SP 2.2-1 no SPEM. ....	72
Figura 25. Representação da SP 3.1-1 no SPEM. ....	73
Figura 26. Representação da SP 3.2-1 no SPEM. ....	75

## ÍNDICE DE TABELAS

Tabela 1. Estereótipos do SPEM. ....	42
Tabela 2. Mapeamento entre Processo X SPEM X CMMI.....	47

# 1 INTRODUÇÃO

## *1.1 CONTEXTO*

A importância da engenharia de software para o sucesso de um projeto de software tem se tornado mais evidente. Áreas como Qualidade têm se tornado grandes parceiras na melhoria de processos de software e aumento da produtividade das empresas de desenvolvimento visto que grande parte dos projetos não consegue entregar os produtos dentro dos padrões de qualidade, de cronograma e de custo estimados.

A inserção de processos de software com metodologias, procedimento e práticas para a melhoria da qualidade e produtividade do desenvolvimento de sistemas vêm se tornando um setor de mais investimento em organizações que desejam melhorar sua competitividade no mercado. [5]

A aquisição de certificações em processo de softwares como padrão ISO 9000-3 tem agregado valor competitivo as organizações essencialmente no panorama nacional e tem a finalidade de padronizar o processo, reduzindo custos.

Apesar da ISO 9000-3 possuir uma boa aceitação no cenário brasileiro, empresas de tecnologia que desejam uma visibilidade internacional têm aderido a normas mais conceituadas e já difundidas no mercado como o CMM (Capability Maturity Model). Porém este modelo está entrando em descontinuidade devido a uma versão mais completa e robusta motivada pela composição de várias práticas, modelos e normas existentes, intitulado CMMI (Capability Maturity Model Integrated).

Esta gama de modelos de processo torna difícil avaliar a aderência e aspectos relevantes entre diferentes processos institucionalizados, o que poderia contribuir de forma significativa a melhoria dos processos como dos próprios modelos.

## ***1.2 OBJETIVOS***

Em busca de contribuir na melhoria do processo de software, surgiu a idéia de promover o desenvolvimento de um ambiente de elaboração de processos desde sua concepção até a instanciação em um estudo de caso de projeto. O ambiente visa a adaptação do meta-modelo de processo de software<sup>1</sup> para cada projeto, apresentando sugestões, tanto de modelo de ciclo de vida, quanto de atividades, procedimentos e ferramentas, entre outros.

Um outro atrativo do ambiente é a possibilidade de personalizar e aumentar a base de históricos de execução de projetos (a partir da definição de processo de software), dessa forma, a qualidade e o grau de acerto das sugestões do sistema têm uma otimização significativa, pois a medida que o produto de software vai sendo desenvolvido, uma visão mais precisa das necessidades reais do projeto, sejam ferramentas, documentos ou mesmo treinamentos, fica mais evidente.

A **Figura 1**, mostra um esquema do modelo utilizado pelo Ambiente na adaptação do meta-modelo de processo, composto por três atividades

---

<sup>1</sup> Componente (*framework*) definido para unificar as terminologias das características (atividades, artefatos, perfis, etc.) que definem um processo de software, baseado em modelos/normas de qualidade para processo de software.

principais: Definição do Processo Padrão, Especialização do Processo e Instanciação do Processo.

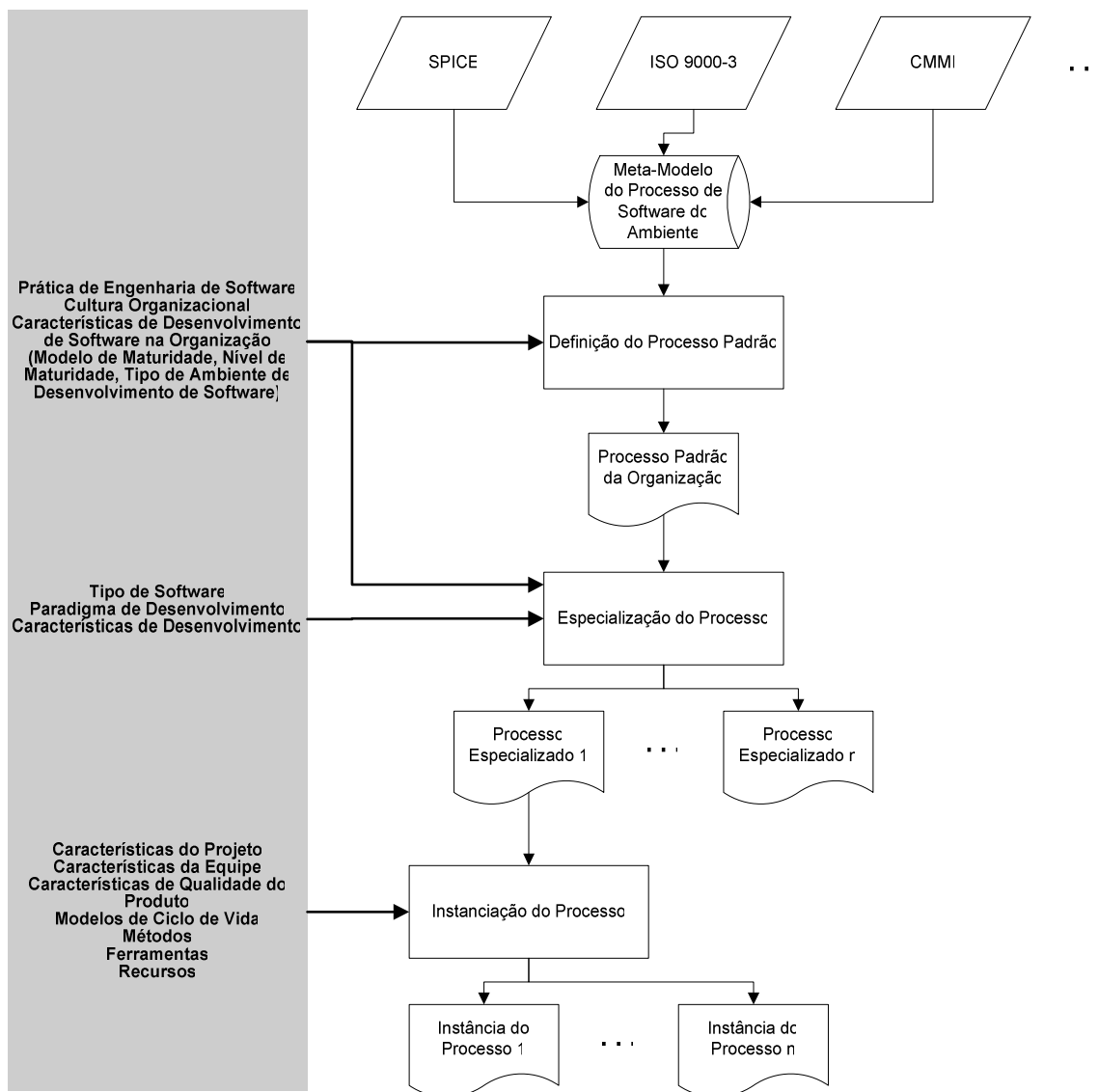


Figura 1. Esquema utilizado pelo ambiente na adaptação do meta-modelo de processo.

Na estrutura definida na **Figura 1**, está representado o Meta-modelo de processo de software, composto de componentes e dos relacionamentos entre esses com influências de algumas normas e modelos de qualidade para processo de software (CMMI, SPICE – ISO 15504 [ISO 1998], ISO 9000–3 [ISO 1997]). O objetivo deste meta-modelo

é determinar uma terminologia única para a definição de processos de software no Ambiente.

A definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e constitui a base para a definição de todos os seus processos. Dessa forma, estabelece-se um processo básico que servirá como ponto de partida para a posterior definição dos processos de software adequados às diferentes características de cada projeto, permitindo economia de tempo e esforço na definição de novos processos. Nesse modelo, a definição do processo padrão é realizada tendo como base o meta-modelo de processo de software e as características do desenvolvimento de software na organização.

Tendo em vista que tipos de software diferentes possuem características distintas e requerem diferentes abordagens de desenvolvimento, o processo de software padrão da organização deverá ser adaptado (especializado) considerando-se as características relacionadas ao tipo de software (por exemplo, sistemas de informação) e ao paradigma de desenvolvimento utilizado (por exemplo, orientação a objetos). Assim, durante a etapa de especialização do processo padrão, atividades poderão ser adicionadas ou modificadas, de acordo com o contexto para qual se está realizando a especialização.

A instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades. Nesta etapa, são definidos o modelo de ciclo de vida, os métodos e as ferramentas que serão utilizadas no projeto, os recursos

humanos e suas responsabilidades ao longo do processo e os artefatos (produtos) consumidos e gerados. [7]

O objetivo deste trabalho é contribuir para a definição deste meta-processo, fazendo uma avaliação da aderência do modelo proposto pelo CMMI ao modelo SPEM.

Através do mapeamento entre os componentes é possível verificar quais os pontos em que os modelos têm componentes semanticamente parecidos e outros que, para uma melhora aderência, requerem alguma condição, restrição ou até mesmo composição de mais de um componente do modelo.

Através dos estudos de caso é possível ter um cenário mais prático de como os modelos se relacionam, permitindo analisar o grau de aderência e identificar componentes que não possuem equivalente em outro modelo.

A partir dos resultados deste trabalho é possível extrair informações relevantes para uma proposta que satisfaça as principais práticas e normas da engenharia de software no âmbito de um modelo para processos de softwares, facilitando a proposição de ferramentas para extração de métricas e desenvolvimento de sistemas e aplicações, como o ambiente proposto.

### *1.3 ESTRUTURA DO TRABALHO*

Além deste capítulo introdutório, este trabalho é composto também pelos seguintes capítulos:



- Capítulo 2 – O modelo CMMI, que define os principais componentes que formam o modelo, fazendo uma alusão à história e motivação de sua criação;
- Capítulo 3 – O Modelo SPEM, faz uma breve descrição dos componentes que formam o modelo e definindo seus estereótipos;
- Capítulo 4 – Mapeamento CMMI e SPEM no Processo, explica sobre o mapeamento dos componentes do CMMI no SPEM, tendo como referências ontologias de um processo de software;
- Capítulo 5 – Em Estudos de Caso, foi feito o mapeamento duas áreas de processo do CMMI – Gerência de Requisitos e Gerência de Configuração – em SPEM;
- Capítulo 6 – Conclusão e trabalhos futuros, neste último estarão sendo feitos o fechamento do trabalho, as conclusões e as perspectivas para trabalhos futuros.

## 2 O MODELO CMMI

### 2.1 CAPABILITY MATURITY MODELS (CMMS)

O SEI (Software Engineering Institute) definiu algumas dimensões em que uma organização que deseja melhorar seu desempenho pode focar. A **Figura 2** demonstra essas dimensões críticas e podem ser agrupadas em três: pessoas, procedimentos e métodos e ferramentas e equipamentos.

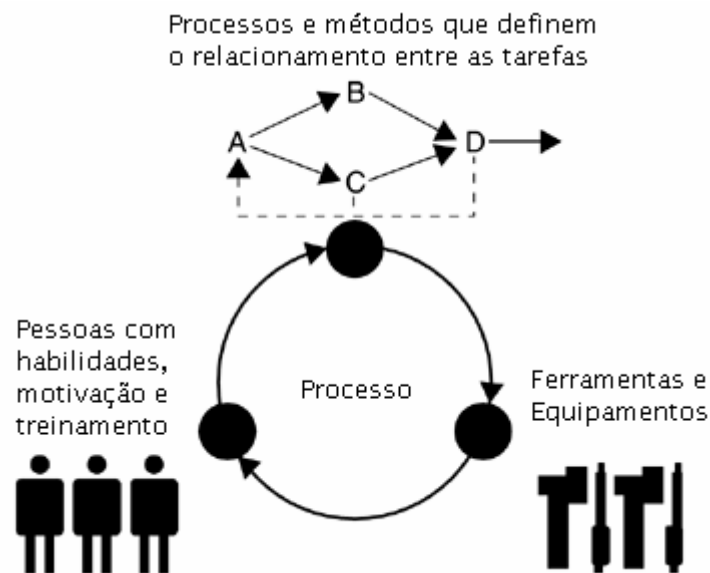


Figura 2. As três dimensões críticas.

A forma com que as dimensões interagem depende do processo utilizado em sua organização. Eles permitem mapear a escalabilidade e fornece um conjunto de práticas de como aperfeiçoar a execução e controle das tarefas.

A importância de ter um processo eficiente, mas principalmente com qualidade tem se tornado mais evidente.

Em meados de 1930, Walter Shewhart iniciou um trabalho que tinha como objetivo a melhoria do processo baseados em princípios de métricas e controle de qualidade. Tais princípios foram refinados por W. Edwards Deming e Joseph Juran. Watts Humphrey, Ron Radice e outros estenderam o modelo existente e começaram a aplicá-lo na produção de softwares na IBM e SEI.

O SEI promoveu a premissa de gerenciamento de processos, “A qualidade de um sistema ou produto está diretamente relacionada a qualidade do processo utilizado para desenvolver ou mantê-lo”. A partir dessa filosofia foram concebidos os modelos de capacitação para maturidade. Esta premissa é presente em todas as iniciativas que promovem a qualidade no mundo inteiro e é incentivada pela Internacional Organization for Standardization/Internacional Eletrotechnical Comission (ISO/IEC). [2]

Um modelo descreve uma representação simplificada do mundo. Capability Maturity Models (CMM) possuem os elementos necessários para modelar processos ou áreas de conhecimento relacionadas.

Desde 1991, CMMs foram desenvolvidos por várias disciplinas. Disciplinas como Engenharia de Sistemas, Engenharia de Software, Aquisição de Software, Gerenciamento de Projeto e Desenvolvimento e Produtos Integrados e Processos de Desenvolvimento. Embora úteis para a maioria das organizações, essa gama de disciplinas se tornou algo de difícil gerenciamento devido a algumas diferenças entre elas, incluindo arquitetura, conteúdo e abordagem, gerando algumas restrições ao tentar aperfeiçoar o desempenho da organização de forma satisfatória.

## *2.2 CMMI – CMM INTEGRATION*

O Projeto CMMI (CMM Integration) foi patrocinado pelo Departamento de Defesa dos Estados Unidos, mais especificamente pelo Escritório de Aquisição, Tecnologia e Logística. Organizações da indústria, governo e do Instituto de Engenharia de Software (SEI) contribuíram na produção do framework CMMI, um conjunto de modelos integrados, de métodos de avaliação do CMMI e com suporte a produtos.

O CMMI foi projetado para solucionar tal deficiência. Baseado em diferentes áreas de conhecimento, foi concebido essencialmente a partir da combinação de três modelos: Capability Maturity Model for Software (SW-CMM) versão rascunho 2.0 C, Electronic Industries Alliance Interim Standard (EIA/IS) 731, e Integrated Product Development Capability Maturity Model (IPD-CMM) versão 0.98.

A confecção do framework a partir da integração de modelos não foi feito de forma simples. Os projetistas do CMMI conceberam um ambiente aderente a múltiplas disciplinas e flexível o suficiente para possuir duas representações: a contínua e a por estágios. [1]

O CMMI é constituído das melhores práticas que estão relacionadas ao desenvolvimento e manutenção de produtos e serviços, fornecendo suporte ao ciclo de vida dos produtos desde a sua concepção até a entrega e manutenção.

Um produto pode ser um avião, um componente de vídeo game, um sistema de controle de mísseis ou um pacote de softwares disponível para um usuário comum. O setor de serviços como turmas de treinamento,

suporte técnico para um determinado software também são suportados pelo modelo CMMI. [2]

### *2.3 DISCIPLINAS (ÁREAS DE CONHECIMENTO) DO CMMI*

O objetivo do CMMI é fornecer um CMM que tenha aborde o desenvolvimento do produto ou serviço além da manutenção, mas que permite a inclusão de novas áreas de conhecimento, que são conhecidas como disciplinas do CMMI. Atualmente o CMMI aborda quatro áreas de conhecimento que auxiliam na melhoria do processo: engenharia de sistemas, engenharia de software, produtos integrados e desenvolvimento de processos e fornecimento de recursos.

#### *2.3.1 ENGENHARIA DE SISTEMAS*

A engenharia de sistemas aborda o desenvolvimento de sistemas completos, que podem ou não incluir software. O enfoque dessa disciplina é capturar as necessidades do cliente, expectativas e restrições em produtos, fornecendo suporte necessário durante toda a vida do produto.

#### *2.3.2 ENGENHARIA DE SOFTWARE*

A engenharia de software aborda o desenvolvimento de sistemas essencialmente de software. O papel dos engenheiros de software é aplicar de forma sistemática, disciplinada utilizando abordagens quantificáveis ao desenvolvimento, operação e manutenção do software.

#### *2.3.3 PRODUTOS INTEGRADOS E DESENVOLVIMENTO DE PROCESSOS*

A área de conhecimento de Produtos Integrados e Processo de Desenvolvimento aborda de maneira sistemática o relacionamento e interação dos stakeholders mais representativos durante o tempo de vida

do produto, objetivando a satisfazer as necessidades do cliente, expectativas e requisitos. Os processos que contribuem com esta disciplina estão integradas à outros processos na organização.

#### *2.3.4 FORNECIMENTO DE RECURSOS*

A disciplina de Fornecimento de Recursos tem como objetivo abordar a aquisição de produtos que podem melhorar, agilizar ou simplificar o projeto, principalmente quando o esforço de trabalho é muito extenso ou complexo.

### *2.4 TIPOS DE REPRESENTAÇÕES*

O CMMI possui dois tipos de representações: a contínua e a por estágios de maturidade. Cada uma das abordagens possui aspectos positivos e negativos sendo, portanto uma análise de qual se adapta melhor aos interesses da organização.

#### *2.4.1 REPRESENTAÇÃO CONTÍNUA*

A representação contínua oferece uma abordagem mais flexível na melhoria do processo. Uma organização pode escolher aperfeiçoar o desempenho de apenas um ponto deficiente no processo ou pode atacar áreas que estão diretamente alinhadas aos objetivos do negócio da organização.

Níveis de capacitação são utilizados para medir a distância entre cada área de processo e sua respectiva otimizada. Em outras palavras, se uma organização deseja alcançar o nível de maturidade 2 em uma determinada área de processo e em outra o nível 4. A medida que a

organização alcançar o nível pretendido ela pode ampliar seu escopo para as demais áreas de processo.

#### *2.4.2 REPRESENTAÇÃO POR ESTÁGIOS*

A representação por estágios possui uma forma mais sistemática, estruturada de melhorar o processo. Essa forma mais cadenciada garante que a base para um próximo nível de maturidade foi aplicada adequadamente.

As áreas de processo são organizadas por níveis de maturidade e orienta os passos a serem contemplados de um nível de maturidade inicial até um nível otimizado, dessa forma permite que o processo possa ser melhorado de forma incremental.

### *2.5 CATEGORIAS DOS COMPONENTES DE UMA ÁREA DE PROCESSO*

Os componentes de uma área de processo podem ser agrupados em três categorias: requeridos, esperados e informativos.

#### *2.5.1 COMPONENTES REQUERIDOS*

Os componentes requeridos são aqueles considerados obrigatórios para satisfazer uma área de processo. A implementação do componente deve estar visível no processo da organização. No caso do CMMI os componentes requeridos são os Specific Goals (Objetivos Específicos) e os Generic Goals (Objetivos Genéricos). A realização de um objetivo é um item de avaliação para saber se uma área de processo foi alcançada e satisfeita.

### *2.5.2 COMPONENTES ESPERADOS*

Os componentes esperados são geralmente implementados para contribuir na realização dos componentes requeridos. Eles servem como guias para quem implementa as melhorias ou realiza as avaliações. No CMMI os Specific Practices (Práticas Específicas) e os Generic Practices (Práticas Genéricas) representam esse tipo de componente.

### *2.5.3 COMPONENTES INFORMATIVOS*

Os componentes informativos fornecem detalhes que ajudam definir de que forma os componentes requeridos e obrigatórios podem ser implementados. Esses componentes no CMMI são representados por Subpractices (Subpráticas), Typical Work Products (Artefatos), Discipline Amplifications (Amplificações), Generic Practice Elaborations (Elaborações de Práticas Genéricas), os títulos das práticas e objetivos, Goal (Objetivo) e Practice Notes (Notas sobre a Prática) e referências.

## *2.6 COMPONENTES RELACIONADOS*

Os componentes associados ser resumidos no diagrama da **Figura 3**:



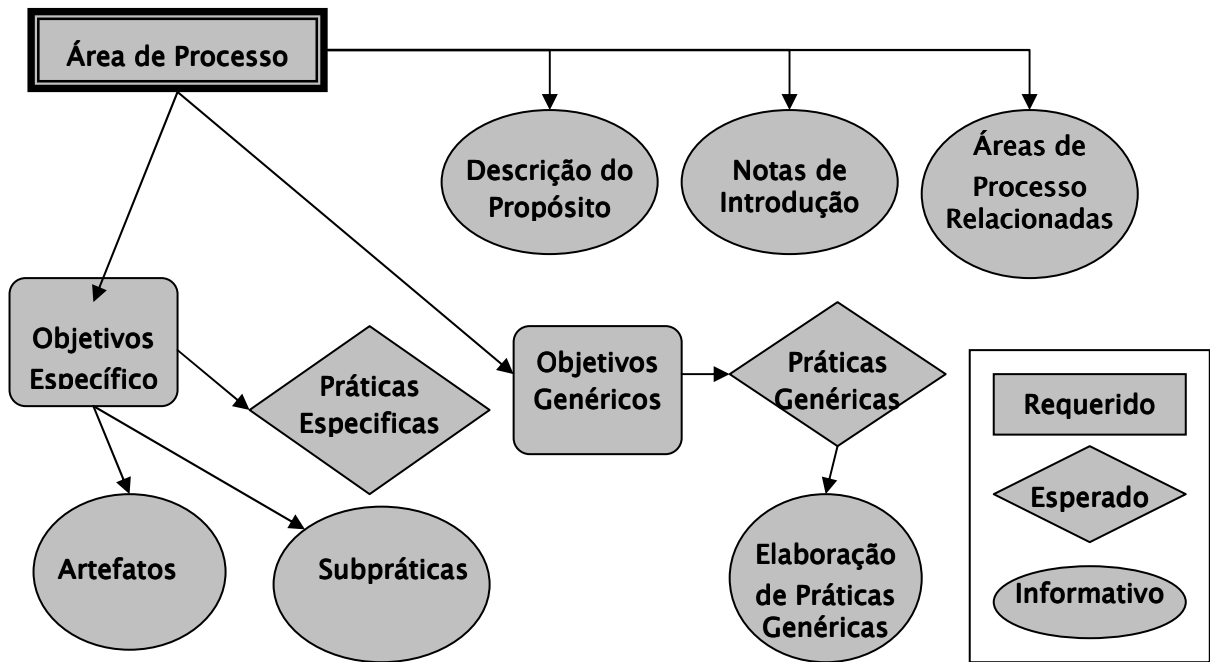


Figura 3. Componentes do Modelo CMMI.

### 2.6.1 PROCESS AREA

Área de processos é uma coleção de práticas relacionadas que, quando implantadas coletivamente, satisfazem um conjunto de objetivos considerados importantes para melhorias na área em questão.

### 2.6.2 PURPOSE STATEMENT

A declaração da finalidade descreve o objetivo da área de processo e é um componente informativo.

### 2.6.3 INTRODUCTORY NOTES

A seção de notas de introdução de uma área de processo descreve os principais conceitos cobertos pela área de processo e é um componente informativo.

#### *2.6.4 RELATED PROCESS AREAS*

A seção de áreas de processo relacionadas lista referências relacionadas à área e reflete um relacionamento de alto-nível entre as áreas de processo. É um componente informativo.

#### *2.6.5 SPECIFIC GOALS*

Um objetivo específico descreve as características que devem ser contempladas para satisfazer a área de processo. É considerado um componente requerido no modelo e é utilizado em avaliações para determinar quais áreas de processo foram satisfeitas.

Somente a definição, do objetivo específico, é considerada um componente requerido no modelo. O título (precedido pelo número do objetivo) e qualquer nota associada ao objetivo são considerados componentes informativos no modelo.

#### *2.6.6 GENERIC GOALS*

Objetivos genéricos aparecem no fim de uma área de processo e são chamados de “genéricos” porque a mesma declaração aparece em múltiplas áreas de processo. Ele descreve as características que devem estar presentes para institucionalizar o processo que implementa a área de processo. É considerado um componente requerido no modelo e é utilizado em avaliações para determinar quais áreas de processo foram satisfeitas.

O título (precedido pelo número do objetivo) e qualquer nota associada ao objetivo são considerados componentes informativos no modelo.

### *2.6.7 SPECIFIC PRACTICES*

Uma prática específica é a descrição de uma atividade considerada importante para alcançar o objetivo específico associado. As práticas específicas descrevem as atividades esperadas que resultem no cumprimento dos objetivos específicos de uma área de processo. É um componente esperado no modelo.

O título (precedido pelo número do objetivo) e qualquer nota associada ao objetivo são considerados componentes informativos no modelo.

### *2.6.8 GENERIC PRACTICES*

Práticas genéricas aparecem próximo ao fim de uma área de processo e são chamadas “genéricas” porque a mesma prática aparece em múltiplas áreas de processo. Uma prática genérica é a descrição de uma atividade considerada importante para a realização de objetivos genéricos associados. É considerado um componente esperado no modelo.

O título (precedido pelo número do objetivo) e qualquer nota associada ao objetivo são considerados componentes informativos no modelo.

### *2.6.9 TYPICAL WORK PRODUCTS*

A seção de típicos artefatos produzidos lista exemplos de saída de uma prática específica. Estes exemplos são chamados “típicos artefatos produzidos” porque frequentemente há outros artefatos produzidos, mas não são listados. É considerado um componente informativo do modelo.

#### *2.6.10 SUBPRACTICES*

Uma subprática é uma descrição detalhada que fornece guias para interpretação e implementação de uma prática específica. Podem ser prescrita, mas na verdade são componentes informativos com a função de fornecer sugestões úteis para a melhoria do processo.

#### *2.6.11 GENERIC PRACTICE ELABORATIONS*

Uma elaboração de prática genérica aparece depois da prática genérica em uma área de processo para fornecer guias de como a prática genérica deve ser aplicada. É considerado um componente informativo no modelo.

### *2.7 CONSIDERAÇÕES FINAIS*

Neste Capítulo foi possível ter uma visão sobre a motivação histórica para a criação do CMMI, suas principais características e definições. A descrição dos componentes que compõem o modelo é importante para a análise e mapeamento entre o SPEM e o CMMI.

## 3 O MODELO SPEM

### 3.1 ARQUITETURA DE MODELAGEM OMG

O SPEM se enquadra no modelo definido pela Object Management Group (OMG) que possui uma arquitetura de quatro camadas descrita na

Figura 4.

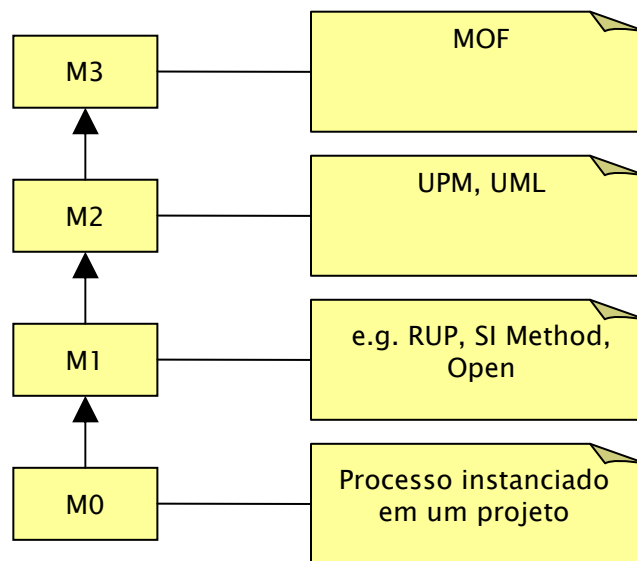


Figura 4. Arquitetura de modelo definido pela OMG.

A camada do M0 define o processo quando ele é instanciando em um projeto. A definição para processos no nível M1 são o RUP (Rational Unified Process), DMR Macroscopic, IBM Global Services Method. O SPEM se encaixa na camada M2, definida como a camada dos metamodelos de processo que servem de template para a camada M1. [3]

A especificação do SPEM (Software Process Engineering Metamodel Specification) é baseada como um tipo de perfil UML, baseado no metamodelo MOF.

### 3.2 O PACOTE SPEM FOUNDATION

O SPEM é um modelo utilizado para especificar, definir processos e seus componentes. O modelo foi construído a partir de um subconjunto, chamado de SPEM Foundation, do metamodelo da UML 1.4. A descreve o pacote SPEM\_Foundation.

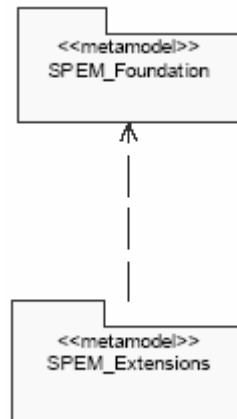


Figura 5. SPEM Foundation: subconjunto da UML 1.4.

#### 3.2.1 SUB-PACOTES DO SPEM\_FOUNDATION

O pacote SPEM\_Foundation é dividido em seis subpacotes:

- **SPEM\_Foundation::Data\_Types** – É um subconjunto do pacote Data\_Types da UML 1.4 e contém as definições dos seguintes tipos de dados: Integer, UnlimitedInteger, String, AggregationKind, Boolean, Boolean, ParameterDirectionKind, PseudoStateKind, Name, Multiplicity e MultiplicityRange. O pacote Data\_Types também possui as definições de Expression e BooleanExpression.
- **SPEM\_Foundation::Core** – Este pacote é estruturado de forma similar ao pacote Core da UML 1.4 e contém os elementos de modelagem que dão suporte a base estrutural do meta-

modelo. Possui elementos que definem relacionamentos, dependências e outros de suporte. Alguns desses elementos são: para relacionamentos Association, AssociationEnd, para dependência Dependency e de suporte ModelElement, PresentationElement.

- **SPEM\_Foundation::Actions** – Este pacote é um subconjunto do pacote Common\_Behaviour da UML 1.4 e define os elementos Action, CallAction e Operation;
- **SPEM\_Foundation::State\_Machines** – Este pacote é um subconjunto do pacote State\_Machines da UML 1.4 e define elementos que representam o conceito de máquina de estado como Transition, State, Action, etc;
- **SPEM\_Foundation::Activity\_Graphs** – Este pacote é um subconjunto do pacote Activity\_Graphs da UML 1.4 e define elementos de modelagem de diagrama de atividades como State, Classifier, etc; [4]
- **SPEM\_Foundation::Model\_Management** – Este pacote é um subconjunto do pacote Model\_Management e os elementos deste pacote são definidos exatamente como na especificação da UML 1.4. e mostra que em SPEM todos os elementos têm visibilidade pública e que os elementos importados para os pacotes não podem ser renomeados.

### ***3.3 MODELO CONCEITUAL***

Nos fundamentos do SPEM, um desenvolvimento de processo de software é uma colaboração entre atividades ativas abstratas chamadas

*Process Roles* que executam operações chamadas *activities* em entidades tangíveis e concretas chamadas *work products*. A Figura 6 descreve um exemplo deste modelo conceitual, utilizando a notação UML, para uma classe.

Role
activity1(WorkProduct1)
activity2(WorkProduct2)

Figura 6. Modelo Conceitual de uma classe.

Múltiplos papéis interagem ou colaboram através da troca de *work products* e iniciam a execução ou realização de determinadas atividades. O objetivo principal do processo é levar um conjunto de *work products* a um estado bem definido.

### 3.4 ESTRUTURA DE PACOTES

O SPEM foi construído a partir do pacote SPEM\_Foundation, um subconjunto da UML 1.4. O Pacote SPEM\_Extensions, adiciona a forma de construir e a semântica necessária em um processo de engenharia de software. A Figura 7 mostra como o pacote SPEM\_Extensions é estruturado internamente, em termos de subpacotes e os relacionamentos entre eles.



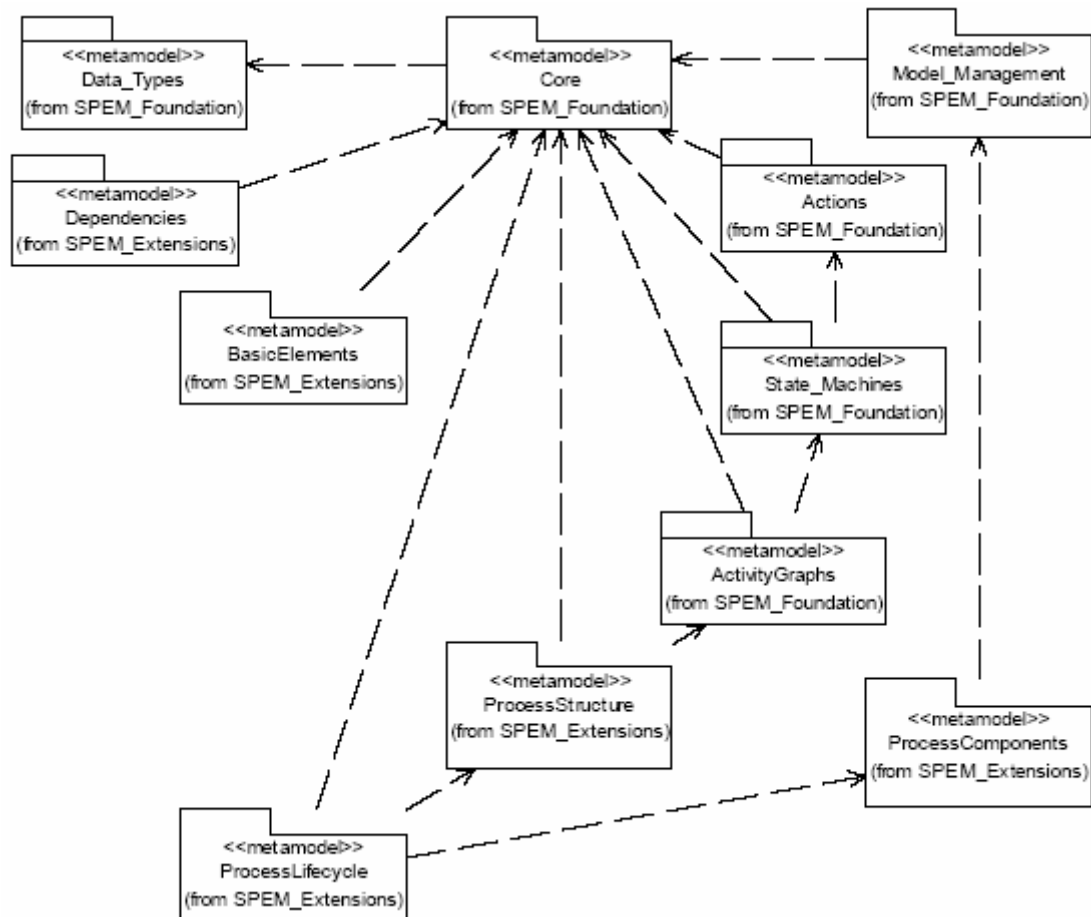


Figura 7. Estrutura do Pacote SPEM.

### 3.4.1 BASIC ELEMENTS (ELEMENTOS BÁSICOS)

A Figura 8 descreve a forma de como os subpacote do pacote Basic Elements se relacionam.

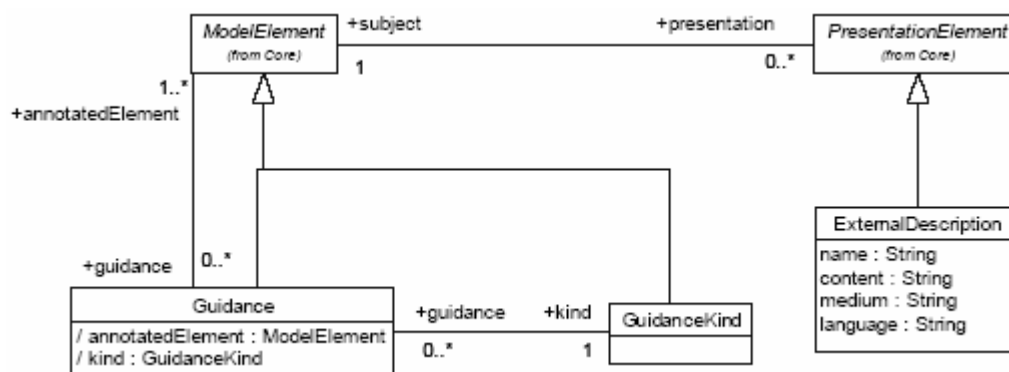


Figura 8. Pacote de Elementos Básicos.

#### 3.4.1.1 ExternalDescription

Contém a descrição de um ModelElement, o que é conveniente para quem vai ler sobre a descrição do processo. ExternalDescriptions compreendem a visão para o usuário da Descrição do Processo de Software.

#### 3.4.1.2 Guidance

São associados aos ModelElements, fornecendo informações mais detalhadas a respeito do ModelElement associado. Os tipos de Guidance dependem da família de processo e podem ser, por exemplo: Guidelines, Techniques, Metrics, Examples, UML Profiles, Tool Mentors, Checklist, Template.

**Technique** é um “algoritmo” detalhado e preciso usado para criar um work product e auxilia na definição das habilidades necessárias para realizar tipos específicos de atividade.

**UMLProfile** fornece mecanismos para especializar a notação UML em análise, projeto por exemplo. Perfis UML como UML para Java, C++, auxiliam a direcionar a execução de uma atividade.

**Checklist** representa um documento que contém uma lista de elementos a serem executados.

**ToolMentor** descreve como utilizar uma ferramenta específica para concluir uma tarefa. Cada ToolMentor é associado a uma ferramenta e estabelece a associação com a Activity por servir de suporte. “Usando o Rational ClearCase para check in check out de itens de Configuração” é um tipo de ToolMentor encontrado no RUP.

**Guideline** é um conjunto de regras e recomendações de como um *work product* deve ser organizado e estruturado. No RUP *Java*

*Programming Guidelines* é um exemplo de guideline para a implementação e projeto de classes, além de ser um item de entrada para a atividade de revisão de código.

**Template** são documentos predefinidos que fornecem um formato padrão para um tipo particular de WorkProduct. “Template para Modelagem de Caso de Uso feito no Microsoft Word” é um exemplo.

**Estimate** decreve o esforço associado a um particular elemento. A descrição associada ao Estimate é responsável por contextualizar e interpretar o esforço.

### 3.4.2 DEPENDENCIES (DEPENDÊNCIAS)

O pacote de Dependencies é definido pelas subclasses do SPEM\_Foundation, que são a Abstraction, Usage e Permission, todas com semântica definida pela UML 1.4. O pacote pode ser visualizado na **Figura 9**.

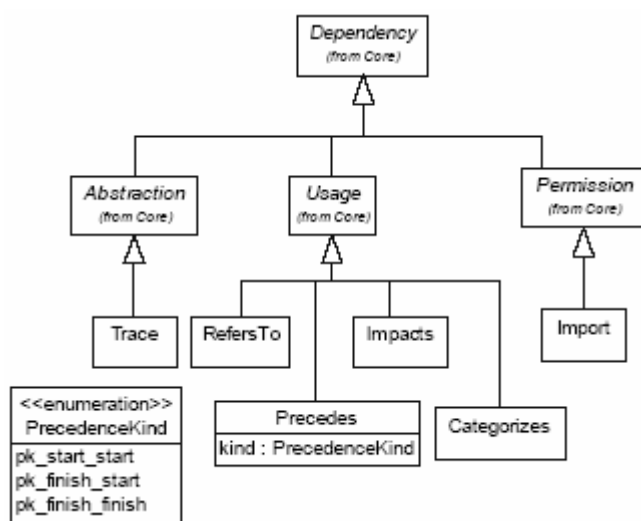


Figura 9. Dependências no SPEM.

#### **3.4.2.1 Categorizes**

Fornece maneiras de associar elementos de processos a múltiplas categorias, atuando de um Package até um elemento de processo de outro pacote. Geralmente é utilizado para categorizar todos os elementos em um alto nível em conjunção com Discipline.

#### **3.4.2.2 Impacts**

Atua entre um WorkProduct e outro WorkProduct para indicar que uma modificação feita em um deles pode invalidar o outro.

#### **3.4.2.3 Import**

Demonstra que o conteúdo do Package destino está no namespace do pacote de origem. Possui semântica parecida com o Import do UML.

#### **3.4.2.4 Precedes**

Indica dependências na forma fim-início ou fim-fim entre o trabalho descrito, atuando entre Activities para representar o conceito de seqüência.

#### **3.4.2.5 RefersTo**

Atua de um elemento de processo a outro, para garantir que eles pertencem ao mesmo ProcessComponent. Geralmente é aplicado quando o texto de um elemento de processo se refere a outro, por nome ou pelo conteúdo, elemento, enfatizando de forma explícita na representação estrutural a relação entre os elementos.

#### **3.4.2.6 Trace**

É utilizado principalmente para rastrear mudanças e requisitos através dos modelos, atuando entre elementos de modelo. Semanticamente parecido ao Trace da UML.

### 3.4.3 PROCESS STRUCTURE (ESTRUTURA DO PROCESSO)

A estrutura principal dos elementos que descrevem processos é construída utilizando este pacote. Seus elementos e os relacionamentos entre eles podem ser visto na **Figura 10**.

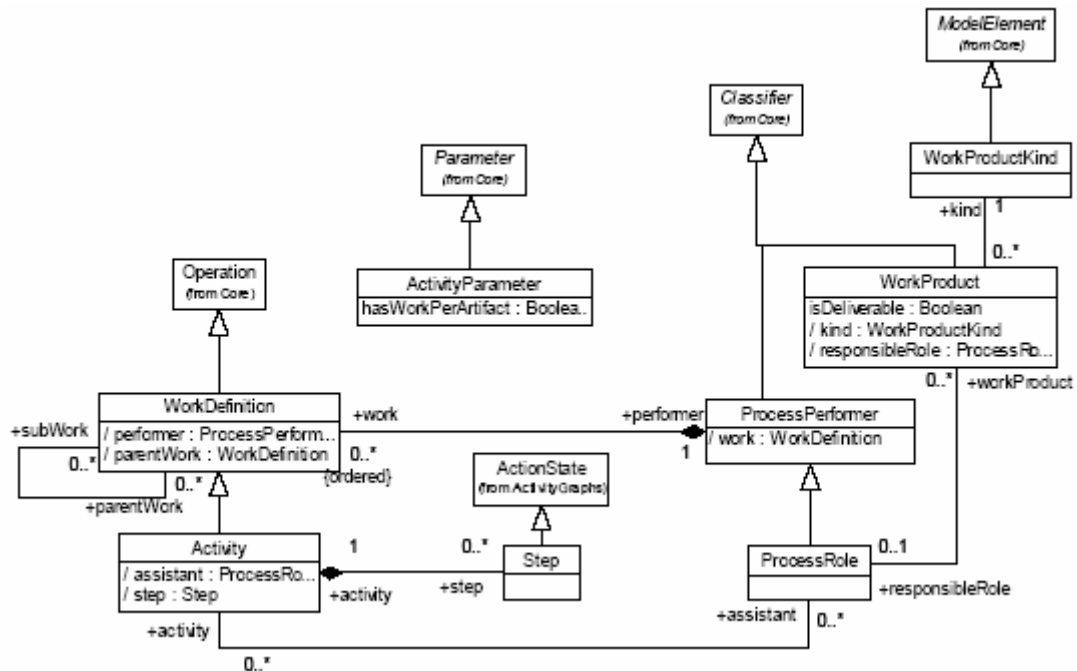


Figura 10. Elementos do pacote Process Structure.

#### 3.4.3.1 WorkProduct e WorkProductKind

Um *work product* (artefato) é qualquer coisa produzida, consumida ou modificada por um processo. Pode ser um documento, um modelo, um código fonte. Um WorkProduct descreve uma classe de artefato produzida no processo.

Um WorkProductKind define o tipo de artefato, se é um documento textual, um modelo UML, um executável.

#### 3.4.3.2 WorkDefinition e ActivityParameter

É um tipo de Operation (operação) que descreve um trabalho executado no processo. Sua principal subclasse é Activity (atividade), mas

Phase (fase), Iteration (iteração) e Lifecycle (ciclo-de-vida) também são WorkDefinitions. Um WorkDefinition pode ser instanciado para representar partes compostas de um trabalho que posteriormente serão decompostas. O ActivityParameter é utilizado para explicitar suas entradas e saídas.

#### **3.4.3.3 Activity (Atividade) e Step (Passo)**

Uma Atividade descreve uma parte do trabalho realizado por um ProcessRole. As tarefas, operações e ações são executadas ou podem ser auxiliadas por um papel. Uma atividade é constituída de elementos atômicos chamados Steps (passos). .

#### **3.4.3.4 ProcessPerformer e ProcessRole**

Um ProcessPerformer define um responsável por realizar um conjunto de WorkDefinitions em um processo. Representa abstratamente o processo como um todo ou um de seus componentes, podendo ser usado para atribuir WorkDefinitions que não possuem um proprietário específico. ProcessRole define responsabilidades para determinados WorkProducts, especificando os papéis que executam e auxiliam atividades específicas.

#### ***3.4.4 PROCESS COMPONENTS (COMPONENTES DE PROCESSO)***

Os elementos desse pacote têm a função de dividir uma ou mais descrições de processo em partes representativas isoladamente que podem ser substituídas através de uma gerência de configuração e controle de mudanças. O pacote pode ser visualizado na **Figura 1**.

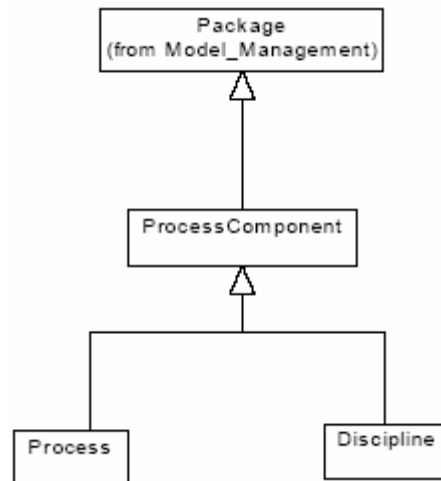


Figura 11. Pacote de Componentes de Processo.

#### 3.4.4.1 Package (Pacote)

Um Package (pacote) é um container que pode tanto possuir como importar elementos de definição de processos, semelhante a UML. Activities (atividades) e WorkDefinitions pertencem a ProcessRoles e ProcessPerformers respectivamente. StateMachines pertencem a WorkProducts (artefatos) e possui seus estados e transições. ActivityGraphs podem pertencer a pacotes, Classifiers (classificadores) ou BehavioralFeatures e um ModelElement do SPEM pode pertencer a um pacote.

Pacotes e a dependência Categorizes podem ser utilizadas para implementar uma categorização genérica para elementos de descrição de processo. Um pacote é criado para representar cada categoria e todos os elementos são relacionados aos pacotes através do Categorizes.

#### 3.4.4.2 ProcessComponent

É um conjunto de descrições de processo que podem ser utilizados por outros ProcessComponents para montar um processo completo. Ele

importa um conjunto de elementos de definições de processos modelados no SPEM por ModelElements.

#### **3.4.4.3 Process (Processo)**

Um processo é um ProcessComponent mas atomicamente completo, ou seja, não pode composto com outros componentes, o que o diferencia de um ProcessComponent convencional. A classe Process pode também representar uma família de processos, em que os processos podem ser redefinidos.

#### **3.4.4.4 Discipline (Disciplina)**

Uma disciplina é uma especialização particular de pacote que agrupa as atividades de processo de acordo com um tema em comum. Esse tipo de separação implica que os Guidances e os artefatos devem ser também categorizados por tema. A inclusão de uma atividade em uma disciplina é representada pela dependência Categorizes, com a restrição de que cada atividade só é categorizada por uma disciplina.

#### ***3.4.5 PROCESS LIFE CYCLE (CICLO DE VIDA DO PROCESSO)***

Os elementos de descrição do processo desse pacote são utilizados para definir como os processos serão executados. Eles descrevem ou restringem o comportamento de um processo em execução, e são utilizados no planejamento, execução e monitoramento do processo. A estrutura deste pacote pode ser visualizada na **Figura 12**.



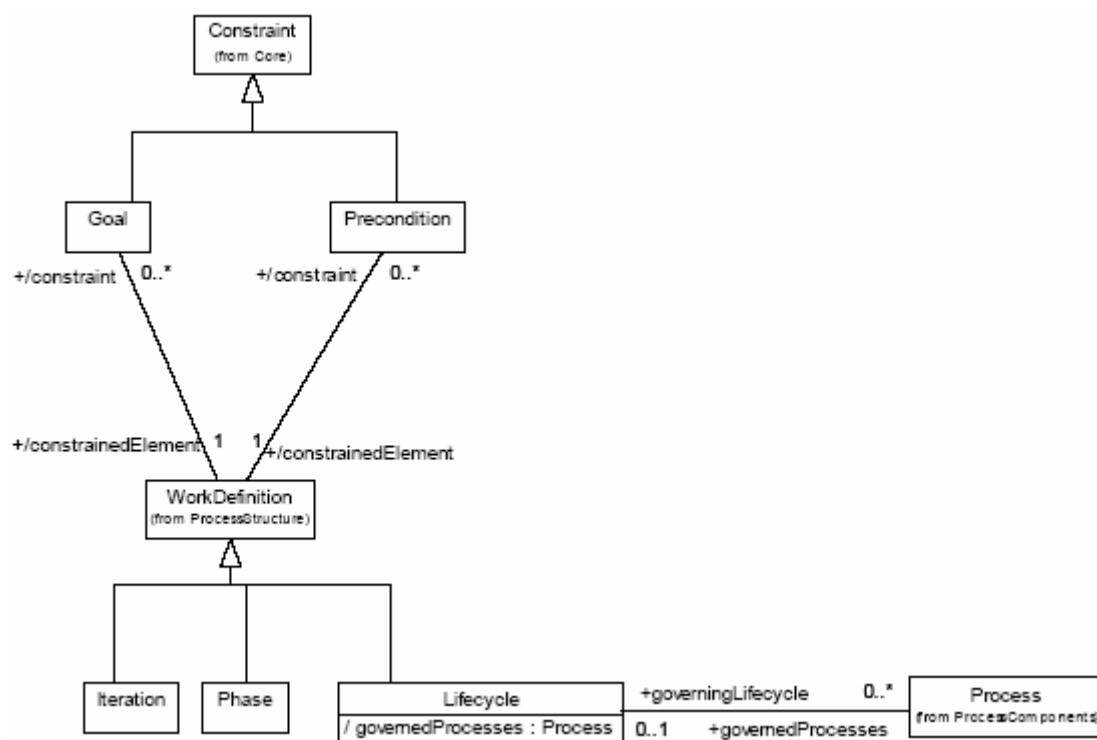


Figura 12. Pacote de Ciclo de vida do Processo.

#### 3.4.5.1 Phase (Fase)

Uma fase é uma especialização de WorkDefinition em que uma precondição define seu critério de entrada e seu objetivo (geralmente chamado de “milestone” ou “marco”) define o critério de saída.

#### 3.4.5.2 Lifecycle (Ciclo de vida)

Um ciclo de vida é definido com uma seqüência de fases que tem um objetivo específico, definindo o comportamento de todo o processo a ser desempenhado em um projeto ou programa.

#### 3.4.5.3 Lifecycle (Ciclo de vida)

É um WorkDefinition composto que possui um milestone (marco) de pequeno porte.


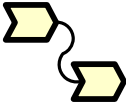


#### 3.4.5.4 Precondition (Pré-condição) e Goal (Objetivo)




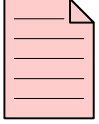
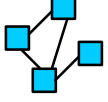

Uma *WorkDefinition* pode ser associada a uma pré-condição e a um objetivo. As precondições e os objetivos são Constraints (restrições), em que a restrição é expressa na forma de uma *BooleanExpression* (expressão booleana) com uma sintaxe similar a uma condição na UML.

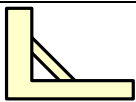
### 3.5 ESTEREÓTIPOS DO SPEM PROFILE

A linguagem SPEM oferece algumas representações e estereótipos para modelar seus principais elementos em diagramas UML. Um resumo dos principais elementos de SPEM, seus conceitos e suas representações gráficas são mostrados na **Tabela 1**.

Tabela 1. Estereótipos do SPEM.

 <i>WorkProduct</i>	Um artefato é qualquer coisa produzida, consumida, ou modificada por um processo. Pode ser um trecho de uma informação, um documento, um modelo, um código fonte, e assim por diante. <i>WorkProduct</i> descreve uma classe de artefato produzida no processo.
 <i>WorkDefinition</i>	<i>WorkDefinition</i> é um tipo de operação que descreve o trabalho realizado no processo. Sua principal subclasse é Atividade, mas Fase, Iteração e Ciclo-de-Vida (no pacote de Processo de Ciclo-de-Vida) também são subclasses.
 <i>ProcessPerformer</i>	Um <i>ProcessPerformer</i> define o realizador (executor) de um conjunto de <i>WorkDefinitions</i> no processo. Ele representa abstratamente o processo como um todo ou dos seus componentes, e é usado para dar um responsável para <i>WorkDefinitions</i> que ainda não possuem.
 <i>ProcessRole</i>	Um <i>ProcessRole</i> é uma subclasse de <i>ProcessPerformer</i> . Ela define responsabilidades em relação a <i>WorkProducts</i> específicos e define papeis que realizam e auxiliam atividades específicas.

 <p><i>ProcessPackage</i></p>	<p>Notação especial para pacotes no contexto SPEM.</p>
 <p><i>Phase</i></p>	<p>Uma <i>Phase</i> é uma especialização de um <i>WorkDefinition</i> em que sua pré-condição define a fase de critérios de entrada e seus objetivos (frequentemente chamados de “milestone”) definem a fase de critérios de saída. Fases são definidas com uma restrição adicional de seqüencialidade. Suas atribuições são executadas em uma série de “datas marco” distribuídas no tempo e frequentemente assumem uma sobreposição (ou não) entre as suas atividades ao mesmo tempo.</p>
 <p><i>Process</i></p>	<p>Um Processo é um <i>ProcessComponent</i> que representa um processo completo, em toda sua extensão. É diferenciado de componentes de processo convencionais pelo fato de não ter intenção de ser composto por outros componentes. Em um contexto de ferramenta,, uma instância do <i>Process</i> é a origem do modelo de processo, onde a ferramenta pode iniciar a condição de transição computacional de todo o restante do processo.</p>
 <p><i>Document</i></p>	<p>Notação específica para diferentes tipos de <i>WorkProduct</i>.</p>
 <p><i>UMLModel</i></p>	<p>Notação específica para diferentes tipos de <i>WorkProduct</i>.</p>
 <p><i>Activity</i></p>	<p><i>Activity</i> é a principal subclasse de um <i>WorkDefinition</i>. Ele descreve uma parte do trabalho realizado por um <i>ProcessRole</i>: suas tarefas, operações e ações executadas por um papel ou de que forma o papel deve auxiliar. Uma atividade pode consistir de elementos atômicos chamados <i>Steps</i>.</p>

 <p><i>Guidance</i></p>	<p>Elementos <i>Guidance</i> podem ser associado a <i>ModelElements</i> que fornecem mais informações detalhadas a respeito do <i>ModelElement</i> associado. Alguns dos possíveis tipos de guia dependem da família do processo e podendo ser, por exemplo: Guidelines, Técnicas, Métricas, Exemplos, Perfis de UML, Mentores de Ferramenta, Lista de Checagem, Templates.</p>
--	---

### 3.6 CONSIDERAÇÕES FINAIS

O SPEM, como foi visto, é uma linguagem para modelagem de processos e tem como ponto forte recursos e componentes para modelagens principalmente comportamentais. A definição de cada um componentes que do SPEM\_Foundation facilita análise semântica entre os seus componentes e os do modelo CMMI.

No capítulo a seguir é feita a descrição de um processo utilizado como referência a partir de ontologias, definindo seus componentes. O mapeamento dos componentes do CMMI e do SPEM a partir do processo de referência também é abordado no próximo capítulo.

## 4 MAPEAMENTO CMMI E SPEM NO PROCESSO

A ontologia de processo para desenvolvimento de software utilizada para a especificação do meta-modelo é baseada no trabalho de Falbo [6] e é utilizada como referência para os modelos SPEM e CMMI.

### *4.1 REPRESENTAÇÕES DO PROCESSO*

#### *4.1.1 PROCESSO*

É um conjunto de atividades estruturadas junto com todo o arcabouço necessário para a sua realização (artefatos, procedimentos e recursos).

Basicamente, um processo consiste de um conjunto estruturado de atividades e, por conseguinte, toda a infra-estrutura envolvida na realização destas (artefatos, procedimentos e recursos).

#### *4.1.2 MODELO DE CICLO DE VIDA*

Estrutura atividades e define abordagem de como organizar um projeto em fases. O ciclo de vida é iniciado quando um software é concebido até quando entra em desuso, ou seja, contém um conjunto de atividades de desenvolvimento, operação e manutenção.

#### *4.1.3 COMBINAÇÃO*

Define a forma como um conjunto de fases de um modelo de ciclo de vida deve ser realizado e especificando o tipo de ordenação em que a estrutura pode ser seqüencial ou iterativa.

#### *4.1.4 ATIVIDADE*

O conceito de atividade está presente em todos os modelos de processo de software e são consideradas primitivas que geram artefatos a partir de artefatos de entrada auxiliados por recursos. Atividades podem corresponder a diferentes níveis, seja uma tarefa ou uma etapa do processo de desenvolvimento.

#### *4.1.5 ARTEFATO*

É a entrada ou produto de uma atividade, podendo ser artefatos de código, documentos ou componentes de software.

#### *4.1.6 PROCEDIMENTO*

São utilizados para auxiliar a realização das atividades, podendo ser direcionados a um tipo específico de atividade, devendo ser adequados a uma tecnologia de desenvolvimento e a um paradigma.

#### *4.1.7 RECURSO*

São elementos necessários para a realização de uma atividade, tais como agentes humanos, equipamentos de hardware e ferramentas de software. Apóiam ou atuam na realização da atividade, mas não podem ser considerados “matérias-primas” para a atividade, ou seja, apenas auxiliam o processo, mas não são incorporados ao produto de software sendo considerados recursos para a atividade.

#### *4.1.8 PADRÃO DE ATIVIDADES*

Representa um comportamento em que decomposições de uma atividade têm em comum.

#### ***4.1.9 PARADIGMA DE DESENVOLVIMENTO***

São princípios e conceitos que orientam o desenvolvimento. Ex.: Paradigma Estrutural e orientado a objetos.

#### ***4.1.10 TECNOLOGIA DE DESENVOLVIMENTO***

A tecnologia a ser empregada no desenvolvimento do software. Ex.: tecnologia convencional de processamento de dados, tecnologia de sistemas baseados em conhecimento, etc.

#### ***4.1.11 RESTRIÇÕES***

São limitações, restrições a serem aplicadas no processo.

### ***4.2 MAPEAMENTO PROCESSO X SPEM X CMMI***

A exhibe o mapeamento feito entre os componentes do processo referência, do SPEM e do CMMI.

Tabela 2. Mapeamento entre Processo X SPEM X CMMI.

Processo	SPEM	CMMI
Processo	Process	
	ProcessComponent	
Modelo de Ciclo de Vida	LifeCycle	
	Iteration	
Combinação	Phase	Maturity Levels
	ProcessPackage	Discipline
Atividade	WorkDefinition	Specific Goals
	Discipline	Process Área
	Activity	Specific Practices
	Step	Subpractices

Artefato	WorkProduct Document UMLModel	Typical Work Product
Procedimento	Guidance Guideline Technique UMLProfile ToolMentor CheckList Template	Subpractices Generic Practice Elaboration Shared Vision Amplification
Recurso	ProcessPerformer ProcessRole	Stakeholder Manager
Padrão de Atividades	Step	Specific Practice
Paradigma de Desenvolvimento		
Tecnologia de Desenvolvimento		
Restrições	ExternalDescription Goal Precondition ActivityParameter trace refersTo categorizes precedes impacts import governs assist perform	Generic Goal Generic Practice Purpose Statement Introductory Notes Related Process Areas



A representação Processo no processo de referência foi mapeado em dois componentes no SPEM que possuem uma semântica semelhante que expressam um conjunto estruturado de atividades para realizar: Process e ProcessComponent. O CMMI não possui nenhum componente equivalente, pois é voltado mais para a parte estrutural do modelo do que a parte comportamental.

O Modelo de Ciclo de Vida do processo de referência pode ser representado no SPEM, como a junção de 2 componentes: LifeCycle e Iteration, descrevendo a vida do software desde sua concepção até seu desuso. O CMMI não aborda ciclo de vida no modelo, mas se pode utilizar em composição com diferentes tipos de ciclos de vida.

Combinação é representada por Phase e ProcessPackage no SPEM, são representados pelos níveis de maturidade, correspondendo a fase de maturidade do processo, em composição com ProcessPackage, que representa a que categoria/família a Área de Processo pertence.

Uma atividade no processo de referência pode ser representada de maneiras diferentes no SPEM e no CMMI, dependendo de sua granularidade. Um Specific Goal no CMMI é representado como um WorkDefinition, descrevendo todo o trabalho realizado para alcançar o objetivo. Uma área de processo no CMMI define uma disciplina no SPEM, por pois agrupa práticas de acordo com um tema em comum. Uma Activity no SPEM é considerada uma SpecificPractice no CMMI por representar uma parte do trabalho a ser realizado. Um Step pode ser representado de duas maneiras no CMMI, dependendo do seu grau de complexidade e ou

subpassos. No caso mais simples ele é uma subprática e no caso de requerer subpassos é representada por uma Technique.

Um artefato é considerado um WorkProduct no SPEM e Typical Work Product no CMMI e possui a mesma semântica. No caso do SPEM ele ainda possui componentes especiais para artefatos em padrão de documentos e para modelos em UML.

Um Procedimento são guias ou itens que auxiliam o processo, sendo representados pelo componente Guidance e suas extensões como Guideline, Template no SPEM e no CMMI podem ser subpráticas, Generic Practices, Elaborations, Shared Vision ou Amplifications, dependendo da semântica da descrição destes componentes.

A representação de Recurso no processo de referência é mapeado no SPEM como um ProcessRole ou um ProcessPerformer no SPEM, pois são necessários para a realização das atividades mas não são artefatos, “matérias-primas”. No caso do CMMI são os stakeholders e managers.

Um padrão de atividade é representado como um Step no SPEM como um passo e no CMMI como uma Specific Practice, por representar um comportamento semelhante em decomposições de uma atividade.

Paradigmas e tecnologias de desenvolvimento do processo referência não possuem representação em componentes no SPEM como no CMMI.

As restrições são representadas no SPEM basicamente por components do pacote Dependency\_Foundation e no CMMI podem ser, dependendo do contexto agregado, Generic Goals, Generic Practices,

Purpose Statement, Introductory Notes, Related Process Áreas, por limitarem, restringirem atividades.

### *4.3 CONSIDERAÇÕES FINAIS*

Neste capítulo foi feita a definição do processo que serviu de suporte para o mapeamento dos componentes do CMMI no SPEM. Com esse arcabouço, foi aplicado o resultado deste mapeamento em dois estudos de caso modelando Áreas de Processo (P.A.) do CMMI – Gerencia de Requisitos e Gerência de Configuração – ambas abordadas com detalhes nos componentes que participam de cada P.A. sob o ponto de vista de componentes do SPEM e em alguns casos demonstrando em diagramas utilizando seus estereótipos.

## 5 ESTUDOS DE CASO

Nesses estudos de caso foram avaliadas duas Áreas de Processo do modelo CMMI – Gerência de Requisitos e Gerência de Configuração, verificando sua aderência aos componentes que compõem o SPEM.

Vale ressaltar que as duas Áreas de Processo foram avaliadas somente sob o âmbito do nível de maturidade 2.

### 5.1 GERENCIAMENTO DE REQUISITOS

A Área de Processo (P.A.) de gerência de requisitos pode ser representada no SPEM como a **Figura 13**.



Figura 13. Estrutura da P.A. de Gerência de Requisitos no SPEM.

A gerência de requisitos é considerada no SPEM uma disciplina que pertence ao pacote de processos (ProcessPackage) de Engenharia de Processos e classificada na fase (Phase) do nível de maturidade 2.

O propósito (Purpose no CMMI) desta PA é gerenciar os requisitos de produtos dos projetos e dos componentes e identificar inconsistências entre os requisitos, planos de projeto e artefatos.

Suas nota introdutória é considerada uma ExternalDescription e descreve com mais detalhes a Área de Processo.

As Áreas de Processo relacionadas (Precondition no SPEM) são:

- Desenvolvimento de Requisitos
- Solução Técnica
- Planejamento de Projeto
- Gerência de Configuração
- Monitoramento e Controle de Projeto
- Gerência de Riscos

#### ***5.1.1SG 1. GERENCIAR REQUISITOS***

Esta área de processo possui na representação de estágios no CMMI apenas um objetivo específico – Specific Goal (S.G.) – que é chamado “Gerenciar Requisitos”, sendo este mapeado como um WorkDefinition no SPEM.

Este objetivo específico que tem como objetivo (Goal) manter um conjunto de requisitos atualizados e aprovados durante a realização do projeto. Ele faz referência (RefersTo) as Áreas de Processo de Soluções Técnicas, de Desenvolvimento de Requisitos e de Monitoramento e Controle de Projeto.

#### ***5.1.2SP 1.1–1. OBTER UM ENTENDIMENTO DOS REQUISITOS***

A **Figura 14** mostra a representação gráfica dos componentes que participam desta atividade.

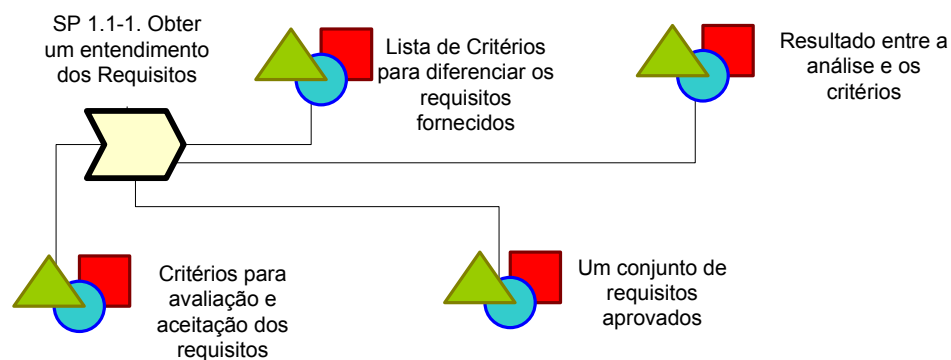


Figura 14. SP 1.1–1 representada no SPEM.

A SP 1.1–1 “Obter um entendimento dos requisitos” é considerada uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) desenvolver um entendimento, juntamente com os fornecedores, do significado dos requisitos.

Nesta prática os seguintes artefatos (WorkProducts no SPEM) participam ou são produzidos:

- Lista de Critérios para diferenciar os requisitos fornecidos
- Resultado entre a análise e os critérios
- Critérios de avaliação e aceitação dos requisitos
- Um conjunto de requisitos aprovados

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Estabelecer critérios para diferenciar os requisitos dos fornecedores. (Step no SPEM)
- Estabelecer critérios objetivos para a aceitação dos requisitos. (Step no SPEM)
- Analisar os requisitos para garantir que os critérios estabelecidos foram alcançados. (Technique no SPEM)

- Obter um entendimento dos requisitos dos fornecedores para que os participantes do projeto possam concordar com eles.  
(Step no SPEM)

### 5.1.3SP 1.2–2. *OBTER O ACORDAMENTO DOS REQUISITOS*

A Figura 15 mostra a representação gráfica dos componentes que participam desta atividade.

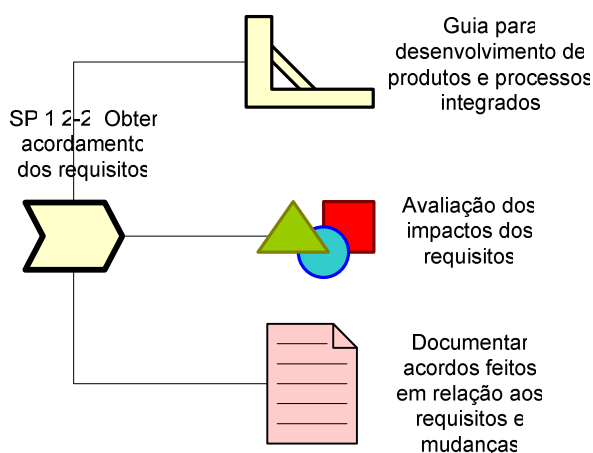


Figura 15. Representação da SP 1.2–2 no SPEM.

Esta prática é considerada uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) que todos os participantes do projeto entrem em acordo em relação aos requisitos.

A amplificação para o caso de se tratar do desenvolvimento de um produto ou processo integrado, que ressalta a importância de que diferentes equipes que participam do projeto também devem acordar com os requisitos, é considerada um Guidance no SPEM.

Os seguintes artefatos (WorkProducts no SPEM) participam ou são produzidos:

- Avaliações dos impactos dos requisitos.

- Documentar acordos feitos em relação aos requisitos e mudanças.

As subpráticas para a realização dessa atividade são:

- Avaliar o impacto dos requisitos no que já foi acordado. (Technique no SPEM)
- Negociar e registrar o que foi acordado. (Technique no SPEM)

#### ***5.1.4SP 1.3-1. GERENCIAR MUDANÇAS NOS REQUISITOS***

A **Figura 16** mostra a representação gráfica dos componentes que participam desta atividade.

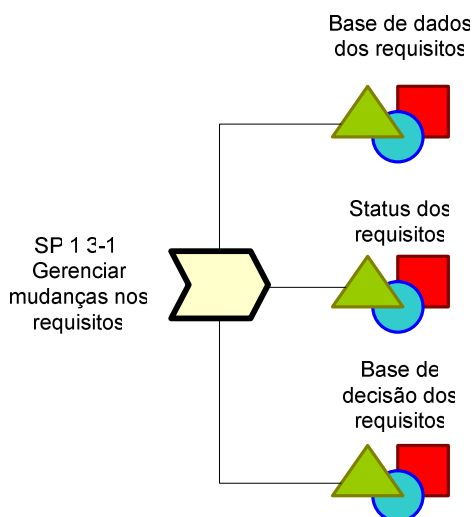


Figura 16. Representação no SPEM da SP 1.3-1.

Esta prática é considerada uma atividade (Activity) no SPEM e é responsável por gerenciar as mudanças ocorridas nos requisitos durante o projeto (Goal no SPEM).

A Área de Processo de Gerência de Configuração é referenciada para fornecer mais detalhes sobre a manutenção e controle dos requisitos sendo, portanto, uma dependência do tipo *RefersTo* no SPEM.



Os artefatos (WorkProducts no SPEM) produzidos ou utilizados nesta P.A. são:

- Status dos Requisitos.
- Base de dados dos requisitos.
- Base de decisão dos requisitos.

As subpráticas para a realização dessa atividade são:

- Capturar todos os requisitos e mudanças ocorridas ou geradas devido ao projeto. (Step no SPEM)
- Manter o histórico de mudanças dos requisitos a motivação da modificação. (Technique no SPEM)
- Avaliar o impacto das mudanças nos requisitos do ponto de vista dos stakeholders mais relevantes. (Technique no SPEM)
- Disponibilizar os dados sobre os requisitos e as mudanças ocorridas disponíveis para o projeto. (Step no SPEM)

#### *5.1.5SP 1.4–2. MANTER A RASTREABILIDADE BIDIRECIONAL DOS REQUISITOS*

A **Figura 17** mostra a representação gráfica dos componentes que participam desta prática específica.

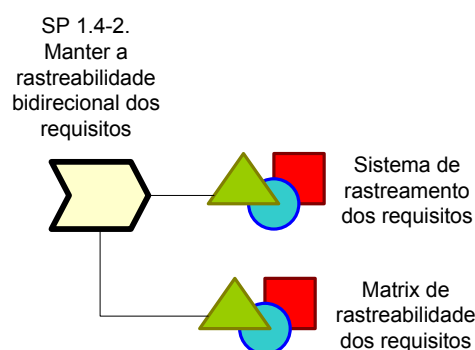


Figura 17. SP 1.4–2 representada no SPEM.

Esta prática é considerada uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) manter o rastreamento bidirecional entre os requisitos, o plano de projeto e os artefatos.

Os artefatos (WorkProducts no SPEM) produzidos ou utilizados nesta P.A. são:

- Matriz de rastreamento dos requisitos.
- Sistema de rastreamento dos requisitos.

Para a realização dessa atividade as subpráticas são:

- Manter a rastreabilidade dos requisitos para garantir que as fontes de baixo nível (derivados) dos requisitos estejam documentadas. (Step no SPEM)
- Manter a rastreabilidade dos requisitos de um requisito até seus derivados e alocações de funções, objetos, pessoas, processos e artefatos. (Step no SPEM)
- Manter a rastreabilidade horizontal de função a função entre as entre interfaces (Step no SPEM).
- Gerar a matriz de rastreabilidade dos requisitos. (Technique no SPEM)

#### *5.1.6SP 1.5–1. IDENTIFICAR INCONSISTÊNCIAS ENTRE O PLANO DE PROJETO E OS REQUISITOS*

A **Figura 18** mostra a representação gráfica dos componentes que participam desta atividade.

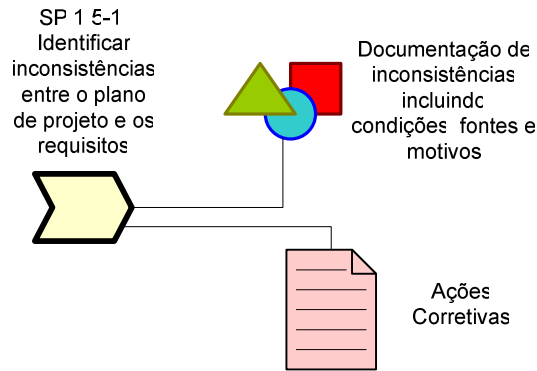


Figura 18. SP 1.5–1 representada no SPEM.

A SP 1.5–1 é considerada uma atividade (Activity) no SPEM e tem como objetivo, Goal no SPEM, identificar inconsistências entre o plano de projeto, artefatos e requisitos.

A Área de Processo de Monitoração e Controle de Projeto é referenciada para fornecer mais detalhes sobre monitoramento e controle de artefatos e planos de projeto, portanto uma dependência do tipo `RefersTo` no SPEM.

Nesta prática os seguintes artefatos (WorkProducts no SPEM) participam ou são produzidos:

- Lista de Critérios para diferenciar os requisitos fornecidos
- Resultado entre a análise e os critérios
- Critérios de avaliação e aceitação dos requisitos
- Um conjunto de requisitos aprovados

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Estabelecer critérios para diferenciar os requisitos dos fornecedores. (Step no SPEM)
- Estabelecer critérios objetivos para a aceitação dos requisitos. (Step no SPEM)

- Analisar os requisitos para garantir que os critérios estabelecidos foram alcançados. (Technique no SPEM)
- Obter um entendimento dos requisitos dos fornecedores para que os participantes do projeto possam concordar com eles. (Step no SPEM)

### *5.1.7GG 2. INSTITUCIONALIZAR PROCESSOS GERENCIADOS*

Um objetivo genérico é considerado no SPEM um componente do tipo pré-condição (Precondition) que deve ser atendido ao se realizar a P.A. de Gerência de Requisitos. E classifica (categorizes no SPEM) o processo como um processo gerenciado.

### *5.1.8GP 2.1. ESTABELECEER UMA POLÍTICA ORGANIZACIONAL*

Uma prática genérica é considerado no SPEM um componente do tipo ActivityParameter pois serve de apoio para realizar os componentes de da Área de Processo. O objetivo (Goal no SPEM) é manter uma política organizacional para planejar e realizar os requisitos do processo gerenciado.

A elaboração é considerada uma técnica (Technique) no SPEM e explica a importância dessa prática para atender as expectativas organizacionais no que se refere ao gerenciamento de requisitos e identificação de inconsistências entre os requisitos, os artefatos e os planos de projeto.

#### *5.1.9GP 2.2. PLANEJAR O PROCESSO*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) estabelecer e manter o plano para a realização dos requisitos em um processo gerenciado.

A elaboração é considerada uma técnica (Technique) no SPEM e explica que o plano de realização dos requisitos é parte do projeto como descrito na Área de Processo de Plano de Projeto (refersTo no SPEM).

#### *5.1.10 GP 2.3. FORNECER RECURSOS*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) fornecer os recursos necessários para execução dos requisitos de um processo gerenciado, desenvolvendo artefatos e fornecendo serviços para o processo.

A elaboração é considerada um ToolMentor no SPEM e afirma que ferramentas para auxiliar o rastreamento dos requisitos são tipos de recursos que podem ser fornecidos.

#### *5.1.11 GP 2.4. ATRIBUIR RESPONSABILIDADE*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) atribuir responsabilidades e autoridades para a realização do processo, desenvolvendo artefatos e fornecendo serviços para o processo de requisitos gerenciado.

#### *5.1.12 GP 2.5. TREINAR PESSOAS*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) treinar e capacitar pessoas para realizar ou dar suporte aos requisitos quando necessário.

A elaboração é considerada um Guideline no SPEM e sugere alguns tópicos de treinamento como, por exemplo, “Definição, análise, revisão e gerenciamento de requisitos”.

#### *5.1.13 GP 2.6. GERENCIAR CONFIGURAÇÕES*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) classificar determinados artefatos dos requisitos de acordo com o nível de gerência de configuração adequado.

A elaboração é considerada um Guideline no SPEM e sugere alguns artefatos que devem ser colocados na gerência de configuração como, por exemplo, requisitos e matriz de rastreabilidade dos requisitos.

#### *5.1.14 GP 2.7. IDENTIFICAR E ENVOLVER OS STAKEHOLDERS MAIS IMPORTANTES*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) identificar e envolver os stakeholder mais importantes no processos de gerência de requisitos quando necessário.

A elaboração é composta por dois componentes do SPEM. A afirmação que afirma que se deve selecionar stakeholder dos clientes, usuários, desenvolvedores e outros que podem afetar o produto ou processo de forma significativa é considerada uma técnica por exigir alguns passos para sua realização. Os exemplos de atividades que o stakeholder pode se envolver é considerada um Guideline no SPEM.

#### *5.1.15 GP 2.8. MONITORAR E CONTROLAR O PROCESSO*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) monitorar e controlar os requisitos de acordo com o plano para a realização de ações corretivas quando necessário.

A elaboração é considerada um Guideline no SPEM e sugere alguns exemplos de métricas que podem ser utilizadas como, por exemplo, a volatilidade dos requisitos.

#### *5.1.16 GP 2.9. OBJETIVAMENTE AVALIAR A ADERÊNCIA*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) avaliar a aderência do processo de gerenciamento dos requisitos em relação a descrição do processo, padrões, procedimentos e não conformidade.

A elaboração é considerada um Guideline no SPEM e sugere alguns exemplos de atividades e artefatos que podem ser revisados como, por exemplo, a atividade de gerência de requisitos e o artefato de matriz de rastreabilidade.

#### *5.1.17 GP 2.10. REVISAR O STATUS COM UM MAIOR NÍVEL DE GERENCIAMENTO*

Considerada no SPEM um componente do tipo ActivityParameter que tem como objetivo (Goal no SPEM) revisar atividades, status e resultados do processo de gerência dos requisitos com um maior nível de gerenciamento e resolver alguns pontos.

A elaboração é considerada uma técnica (Technique) no SPEM e sugere que para garantir que todos os acordos sejam cumpridos, deve-se

aplicar um nível alto de gerenciamento, essencialmente os que foram feitos fora da organização.

## 5.2 GERÊNCIA DE CONFIGURAÇÃO

A Gerência de Configuração é uma Área de Processo (P.A.) do modelo CMMI. Considerada no SPEM uma disciplina que pertence ao pacote de processos (ProcessPackage) de Engenharia de Suporte e classificada na fase (Phase) do nível de maturidade 2. Sua estrutura modelada utilizando estereótipos do SPEM pode ser visualizada na Figura 19.

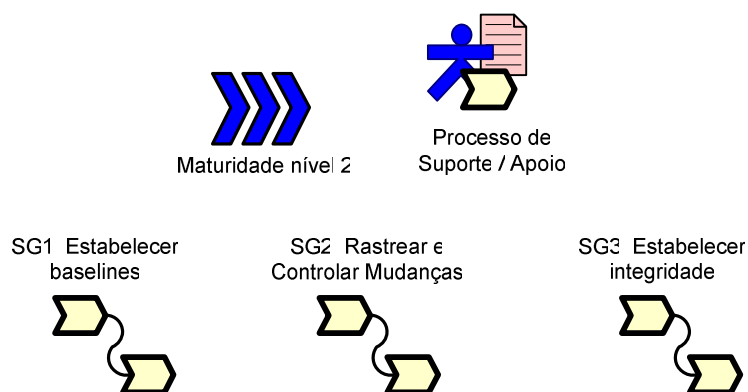


Figura 19. P.A. de Gerência de Configuração nos estereótipos do SPEM.

O propósito (Purpose no CMMI) desta PA é estabelecer e manter a integridade dos artefatos utilizando configurações para identificação, controle, status de contas e auditorias.

Sua nota introdutória é considerada uma ExternalDescription e descreve com mais detalhes a Área de Processo e possui um guia (Guidance) para fornecedores de recursos (*sourcing*).

As Áreas de Processo relacionadas (Precondition no SPEM) são:

- Planejamento de Projeto
- Monitoramento e Controle de Projeto



- Análise de Causa e Resolução

Esta área de processo possui na representação de estágios no CMMI três objetivos específicos, Specific Goal (SG) no SPEM:

- SG 1. Estabelecer baselines
- SG 2. Rastrear e controlar mudanças
- SG 3. Estabelecer integridade

### *5.2.1 SG 1. ESTABELECEER BASELINES*

Este objetivo específico é mapeado como um WorkDefinition no SPEM e tem como objetivo (Goal) estabelecer baselines para os artefatos identificados.

### *5.2.2 SP 1.1–1. IDENTIFICAR OS ITENS DE CONFIGURAÇÃO*

A Figura 20 mostra a representação gráfica dos componentes que participam desta atividade.

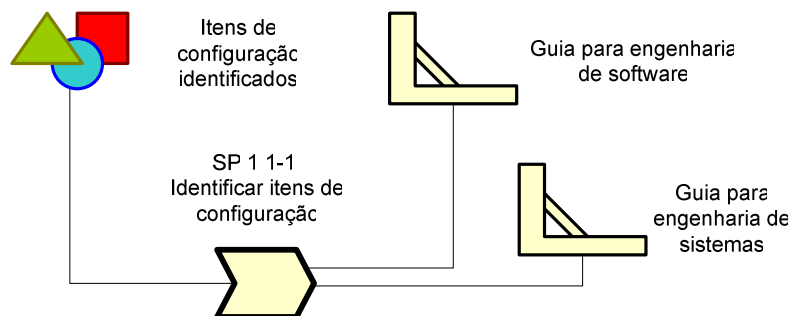


Figura 20. SP 1.1–1 representada no SPEM.

A SP 1.1–1 é considerada uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) identificar os itens de configuração, componentes e artefatos relacionados ao que será colocado sob a gerência de configuração.

A amplificação de Engenharia de Sistemas para o caso de sistemas que incluem hardware e software e que a parte de software é uma pequena parte do sistema, considerada um Guidance no SPEM, sugere todo o software deve ser considerado um item de configuração.

Esta prática possui apenas um artefato (WorkProduct no SPEM):

- Itens de configuração identificados

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Escolher os itens de configuração e os artefatos que os compõem baseado nos critérios documentados. (Step no SPEM) Uma amplificação (Guidance no SPEM) é feita para Engenharia de Software e sugere que artefatos como códigos fonte devem ser considerados itens de configuração.
- Atribuir identificadores únicos aos itens de configuração. (Step no SPEM)
- Especificar as características importantes em cada item de configuração. (Technique no SPEM).
- Especificar quando cada item de configuração é colocado sob a gerência de configuração. (Technique no SPEM)
- Identificar o responsável por cada item de configuração. (Step no SPEM)

#### ***5.2.3SP 1.2–2. ESTABELECE UM SISTEMA DE GERÊNCIA DE CONFIGURAÇÃO***

A Figura 21 mostra a representação gráfica dos componentes que participam desta atividade.

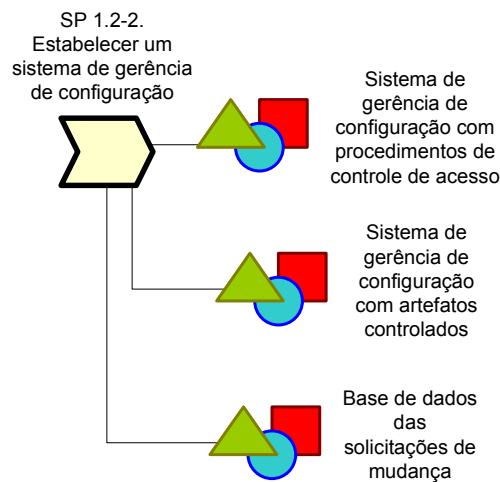


Figura 21. A Prática Específica SP 1.2–2 no SPEM.

A Prática “Estabelecer um sistema de gerência de configuração” é considerada uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) estabelecer e manter uma gerência de configuração e um sistema de gerenciamento de mudanças para artefatos.

Alguns artefatos (WorkProduct no SPEM) participam desta prática:

- Sistema de gerência de configuração com artefatos controlados
- Sistema de gerência de configuração com procedimento de controle de acesso
- Base de dados de solicitações de mudança

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Estabelecer um mecanismo para gerenciar múltiplos níveis de gerência de configuração. (Technique no SPEM)
- Armazenar e recuperar itens de configuração em um sistema de gerência de configuração. (Step no SPEM)

- Compartilhar e transferir itens de configuração entre os níveis de controle no sistema de gerência de configuração. (Step no SPEM)
- Armazenar e recuperar versões arquivadas dos itens de configuração. (Step no SPEM)
- Armazenar, atualizar e recuperar registros de gerência de configuração. (Step no SPEM)
- Criar relatórios de gerência de configuração a partir do sistema de gerência de configuração. (Technique no SPEM)
- Preservar o conteúdo do sistema de gerência de configuração. (Step no SPEM)
- Revisar a estrutura da gerência de configuração quando necessário. (Technique no SPEM)

#### 5.2.4SP 1.3-1. CRIAR OU LIBERAR BASELINES

A Figura 22 mostra a representação gráfica dos componentes que participam desta atividade.

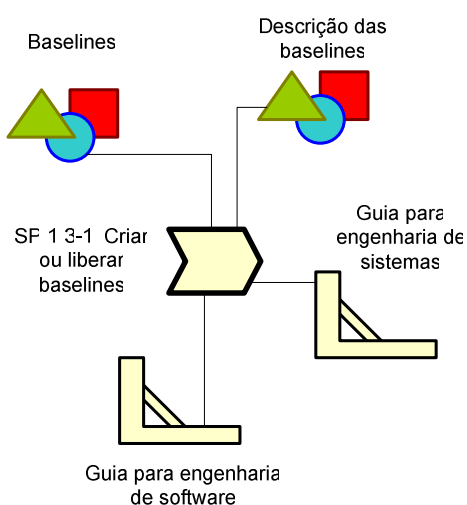


Figura 22. SP 1.3-1 representada no SPEM.

Esta prática, considerada uma atividade (Activity) no SPEM, tem como objetivo (Goal no SPEM) criar ou liberar baselines para uso interno e para uso do cliente.

A amplificação de Engenharia de Sistemas, considerada um Guidance no SPEM, sugere que múltiplas baselines podem ser usadas para definir um produto evolutivo durante seu ciclo de desenvolvimento.

A amplificação de Engenharia de Software, é também uma Guidance no SPEM, explica que uma baseline é composta de um conjunto de requisitos, de projeto, de códigos fontes, documentação do usuário. Quando é feita para o cliente é geralmente é chamada de “*release*” e para uso interno é chamada de “*build*”.

Alguns artefatos (WorkProduct no SPEM) participam desta prática:

- Baselines
- Descrições das baselines

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Obter a autorização dos CCBs (Change Control Board) antes de criar ou liberar baselines dos itens de configuração. (Step no SPEM)
- Criar ou liberar baselines apenas a partir dos itens de configuração do sistema de gerência de configuração. (Step no SPEM) Uma amplificação (Guidance no SPEM) é feita para garantir que os itens de configuração são feitos para o cenário correto.

- Documentar um conjunto de itens de configuração que estão contidos na baseline. (Technique no SPEM)
- Disponibilizar o conjunto de baselines mais atual. (Step no SPEM)

#### 5.2.5SG 2. RASTREAR E CONTROLAR MUDANÇAS

Este objetivo específico é mapeado como um WorkDefinition no SPEM e tem como objetivo (Goal) fazer as mudanças nos artefatos que estão sob a gerência de configuração sejam rastreados e controlados.

#### 5.2.6SP 1.3–1. CRIAR OU LIBERAR BASELINES

A Figura 23 mostra a representação gráfica dos componentes que participam desta atividade.

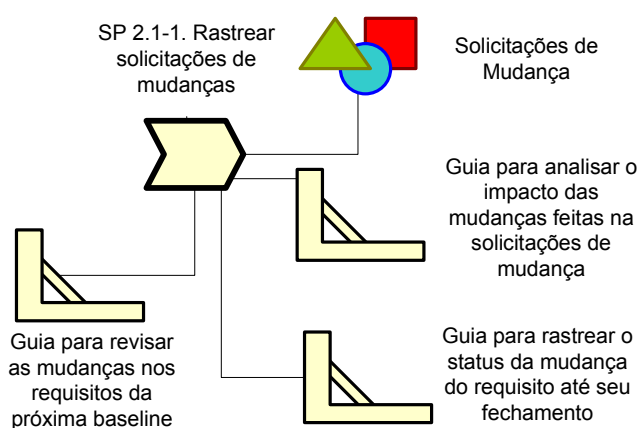


Figura 23. A SP 2.1–1 representada no SPEM.

A SP 2.1–1 é considerada uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) rastrear as solicitações de mudanças feitas para os itens de configuração.

O artefato (WorkProduct no SPEM) que participa desta prática:

- Solicitações de mudança

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Iniciar e registrar a solicitação de mudança na base de dados de solicitações de mudança. (Step no SPEM)
- Analisar o impacto de mudanças e ajustes propostos nas solicitações de mudança. (Technique no SPEM)

Um guia (Guidance no SPEM) ressalta que as mudanças são avaliadas pelo impacto logo após projetar ou contratar requisitos.

- Revisar as solicitações de mudança que participarão da próxima baseline e revisar também os que podem sofrer os efeitos colaterais causados pela mudança. (Technique no SPEM)

Um guia (Guidance no SPEM) descreve que a revisão deve ser feita com os participantes adequados.

- Rastrear o status das solicitações de mudanças para fechamento. (Step no SPEM)

Um guia (Guidance no SPEM) sugere que solicitações de mudanças feitas para o sistema devem ser tratadas de forma eficiente e na hora adequada.

#### ***5.2.7SP 2.2–1. CONTROLE DOS ITENS DE CONFIGURAÇÃO***

A Figura 24 mostra a representação gráfica dos componentes que participam desta atividade.

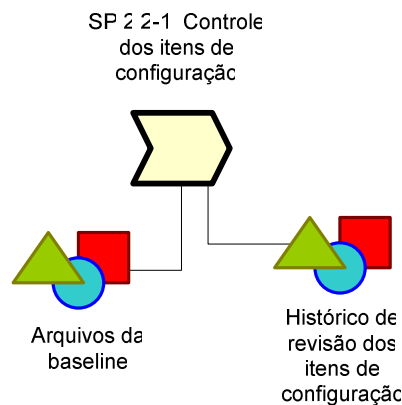


Figura 24. Representação da SP 2.2–1 no SPEM.

A SP 2.2–1 é representada por uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) controlar mudanças dos itens de configuração.

Os artefatos desta prática (WorkProduct no SPEM) são:

- Arquivos da baseline
- Histórico de revisão dos itens de configuração

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Controlar as mudanças nos requisitos durante a vida do produto. (Step no SPEM)
- Obter autorização antes de inserir as mudanças nos itens de configuração no sistema de gerenciamento de mudanças. (Technique no SPEM)
- Realizar o *check in* e *check out* de forma que a corretude e a integridade dos itens de configuração seja mantida. (Technique no SPEM)
- Realizar revisões para garantir que as mudanças não causaram efeitos indesejáveis nas baselines.



- Registrar mudanças nos itens de configuração e, quando necessário, o motivo pelas quais foram feitas.

### 5.2.8SG 3. ESTABELECER INTEGRIDADE

Este objetivo específico é mapeado como um WorkDefinition no SPEM e tem como objetivo (Goal) manter e estabelecer a integridade das baselines.

### 5.2.9SP 3.1-1. ESTABELECER REGISTROS DE GERÊNCIA DE CONFIGURAÇÃO

A Figura 25 mostra a representação gráfica dos componentes que participam desta atividade.

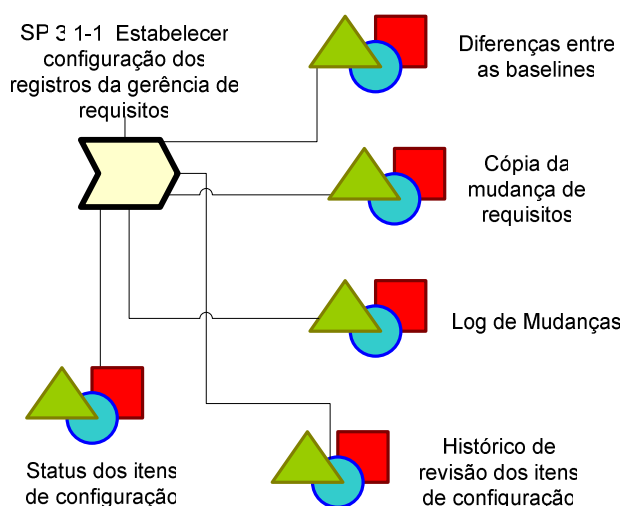


Figura 25. Representação da SP 3.1-1 no SPEM.

Esta prática é representada por uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) estabelecer e manter os registros que descrevem os itens de configuração.

Os artefatos desta prática (WorkProduct no SPEM) são:

- Histórico de revisão dos itens de configuração
- Log de mudanças
- Cópia das solicitações de mudança

- Status dos itens de configuração
- Diferenças entre as baselines

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Registrar as ações feitas na gerência de configuração com detalhes suficientes para o conteúdo e o status de cada item de configuração ser conhecido e poder ser recuperado a partir de versões anteriores. (Step no SPEM)
- Garantir que os *stakeholders* relevantes tenham acesso e conhecimento sobre o status dos itens de configuração. (Technique no SPEM)
- Especificar a versão mais recente das baselines. (Step no SPEM)
- Identificar a versão dos itens de configuração que constituem uma baseline em particular. (Step no SPEM)
- Descrever as diferenças entre baselines sucessivas. (Step no SPEM)
- Revisar o status e o histórico de cada item de configuração quando necessário. (Step no SPEM)

#### **5.2.10 SP 3.2–1. REALIZAR AUDITORIAS DE CONFIGURAÇÃO**

A Figura 26 mostra a representação gráfica dos componentes que participam desta atividade.

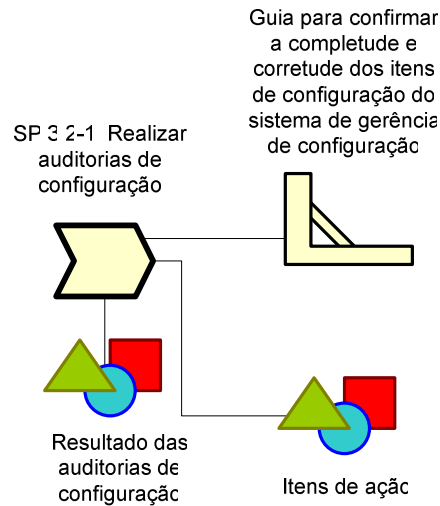


Figura 26. Representação da SP 3.2–1 no SPEM.

Esta prática é representada por uma atividade (Activity) no SPEM e tem como objetivo (Goal no SPEM) realizar auditorias de configuração para manter a integridade de configurações das baselines.

Os artefatos desta prática (WorkProduct no SPEM) são:

- Histórico de revisão dos itens de configuração
- Log de mudanças
- Cópia das solicitações de mudança
- Status dos itens de configuração
- Diferenças entre as baselines

Para a realização dessa atividade, foram estabelecidas algumas subpráticas:

- Avaliar a integridade das baselines. (Technique no SPEM).
- Verificar se os registros de configuração identificam corretamente os itens de configuração. (Step no SPEM)
- Revisar a estrutura e a integridade dos itens de configuração. (Technique no SPEM)

- Verificar a completude e a corretude dos itens de configuração. (Step no SPEM)
- Verificar a aderência dos padrões e procedimentos aplicados na gerência de configuração. (Step no SPEM)
- Rastrear os itens da auditoria até o fechamento. (Step no SPEM)

#### *5.2.11 GG 2. INSTITUCIONALIZAR PROCESSOS GERENCIADOS*

Um objetivo genérico é considerado no SPEM um componente do tipo pré-condição (Precondition) que deve ser atendido ao se realizar a P.A. de Gerência de Requisitos. E classifica (categorizes no SPEM) o processo como um processo gerenciado.

#### *5.2.12 GP 2.1. ESTABELECEER UMA POLÍTICA ORGANIZACIONAL*

Uma prática genérica é considerado no SPEM um componente do tipo ActivityParameter pois serve de apoio para realizar os componentes de da Área de Processo. O objetivo (Goal no SPEM) é manter uma política organizacional para planejar e realizar o processo de gerência de configuração.

A elaboração é considerada uma Guidance no SPEM e explica a importância dessa prática para atender as expectativas organizacionais no que se refere a manter e estabelecer baselines, rastreando e controlando mudanças dos artefatos, preservando a integridade.

#### *5.2.13 GP 2.2. PLANEJAR O PROCESSO*

Considerado no SPEM um componente do tipo ActivityParameter, serve de apoio para realizar os componentes de da Área de Processo. O

objetivo (Goal no SPEM) é estabelecer e manter o plano para realização do processo de gerência de configuração.

A elaboração é considerada Guidance no SPEM e sugere que o plano para realização da gerência de configuração pode ser incluída ou referenciada nos planos de projeto, como descrito na Área de Processo de Planejamento de Projeto (refersTo no SPEM).

#### *5.2.14 GP 2.3. FORNECER RECURSOS*

É considerado no SPEM um componente do tipo ActivityParameter pois serve de apoio para realizar os componentes de da Área de Processo. O objetivo (Goal no SPEM) é fornecer recursos adequados para realização do processo de gerência de configuração desenvolvendo artefatos e fornecendo serviços.

A elaboração é considerada um Guideline no SPEM e sugere alguns exemplos de recursos que podem ser fornecidos como, por exemplo, ferramentas para gerência de configuração.

#### *5.2.15 GP 2.4. ATRIBUIR RESPONSABILIDADES*

Componente do tipo ActivityParameter no SPEM, pois serve de apoio para realizar os componentes de da Área de Processo. O objetivo (Goal no SPEM) é atribuir responsabilidades para a realização do processo, desenvolvendo artefatos e disponibilizando serviços para o processo de gerência de configuração.

#### *5.2.16 GP 2.5. TREINAR PESSOAS*

Considerado no SPEM um componente do tipo ActivityParameter pois serve de apoio para realizar os componentes de da Área de Processo.

Tem como objetivo (Goal no SPEM) treinar as pessoas para realizar ou dar apoio ao processo de gerência de configuração quando necessário.

A elaboração é considerada um Guideline no SPEM e sugere alguns exemplos de tópicos para o treinamento como, por exemplo, “Padrões, procedimentos e métodos para Gerência de Configuração”.

#### ***5.2.17 GP 2.6. GERENCIAR CONFIGURAÇÕES***

Componente do tipo ActivityParameter no SPEM, serve de apoio para realizar os componentes de da Área de Processo. Tem como objetivo (Goal no SPEM) colocar os artefatos de acordo com o nível de gerência de configuração adequado.

A elaboração é considerada um Guideline no SPEM e cita alguns exemplos de artefatos que são colocados sob a gerência de configuração como, por exemplo, listas de acesso.

#### ***5.2.18 GP 2.7. IDENTIFICAR E ENVOLVER OS STAKEHOLDERS RELEVANTES***

Componente do tipo ActivityParameter no SPEM, serve de apoio para realizar os componentes de da Área de Processo. Tem como objetivo (Goal no SPEM) identificar e envolver os *stakeholders* relevantes no processo de gerência de configuração conforme planejado.

A elaboração é considerada um Guideline no SPEM e cita alguns exemplos de atividade que podem ter o envolvimento dos *stakeholders* como, por exemplo, estabelecimentos de baselines.

#### ***5.2.19 GP 2.8. MONITORAR E CONTROLAR O PROCESSO***

É considerado um componente do tipo ActivityParameter no SPEM e serve de apoio para realizar os componentes de da Área de Processo. Tem

como objetivo (Goal no SPEM) monitorar e controlar o processo de gerência de configuração de acordo com o plano para realização do processo e tomar as ações corretivas adequadas.

A elaboração é considerada um Guideline no SPEM e fornece alguns exemplos de métricas utilizadas na monitoria e controle como, por exemplo, número de mudanças ocorridas em um item de configuração.

#### *5.2.20 GP 2.8. MONITORAR E CONTROLAR O PROCESSO*

É considerado um componente do tipo ActivityParameter no SPEM e serve de apoio para realizar os componentes de da Área de Processo. Tem como objetivo (Goal no SPEM) monitorar e controlar o processo de gerência de configuração de acordo com o plano para realização do processo e tomar as ações corretivas adequadas.

A elaboração é considerada um Guideline no SPEM e fornece alguns exemplos de métricas utilizadas na monitoria e controle como, por exemplo, número de mudanças ocorridas em um item de configuração.

#### *5.2.21 GP 2.9. AVALIAR OBJETIVAMENTE A ADERÊNCIA*

É considerado um componente do tipo ActivityParameter no SPEM e serve de apoio para realizar os componentes de da Área de Processo. Tem como objetivo (Goal no SPEM) avaliar de forma objetiva o processo de gerência de configuração em relação a descrição do processo, padrões, procedimento verificando a não conformidade.

A elaboração é considerada um Guideline no SPEM e sugere alguns exemplos de atividades que podem ser revisadas como, por exemplo, rastreamento e controle de mudanças.

### *5.2.22 GP 2.10. REVISAR STATUS EM UM MAIOR NÍVEL DE GERENCIAMENTO*

Considerado um componente do tipo ActivityParameter no SPEM que serve de apoio para realizar os componentes de da Área de Processo. Tem como objetivo (Goal no SPEM) revisar atividades, status e resultados do processo de gerência de configuração com um maior nível de gerenciamento, resolvendo alguns pontos.

## *5.3 CONSIDERAÇÕES FINAIS*

Neste capítulo foi possível ver uma aplicação prática do mapeamento de componentes do CMMI no SPEM.

A aderência dos componentes do CMMI no SPEM que tiveram um mapeamento direto, principalmente devido a semelhança em relação a semântica. São os componentes de objetivos, atividades, artefatos e passos.

Os componentes de mais alto grau como níveis de maturidade e famílias de processo, não tiveram uma aderência completa, notando-se uma necessidade de adicionar restrições e condições para um melhor uso do mapeamento. A falta de componentes que representam notas, dicas encontradas no CMMI não possuem um equivalente semântico adequado no SPEM.

Em suma, o CMMI e o SPEM possuem um grau de semelhança entre componentes razoável, fazendo-se necessário o uso de restrições e condições para melhorar o grau de aderência entre os modelos. Anotações, dicas e observações feitas no CMMI não possuem um componente adequado para ser representado no modelo SPEM, assim como ciclos de vida, encontrados no SPEM, não são cobertos no CMMI.



## 6 TRABALHOS FUTUROS E CONCLUSÃO

### *6.1 TRABALHOS FUTUROS*

Fazer uma avaliação da contribuição que norma ISO-9003 pode agregar a concepção do meta-processo, e compor juntamente com a avaliação dos resultados entre CMMI e o SPEM pode tornar o modelo mais representativo para o ambiente a ser desenvolvido.

Definir o meta-processo de software para o ambiente de desenvolvimento a partir da avaliação dessas normas e modelos e propor um framework que forneça suporte para definição de processos de software em um ambiente de implementação de Processo de Software.

### *6.2 CONCLUSÃO*

Este trabalho teve como objetivo avaliar a aderência entre o modelo CMMI e o SPEM, fazendo um estudo entre os componentes que constituem cada modelo a partir de suas semânticas e aplicações.

O modelo CMMI por ser mais voltado a parte estrutural do processo, possui carências de representação quanto a modelagem de processos com um maior grau de especificação, onde o SPEM possui muitos componentes para representar esse aspecto.

Em contrapartida, os aspectos comportamentais podem ser modelados de forma mais apropriada no SPEM, mas estruturalmente o CMMI é mais representativo.

A carência de componentes do tipo Nota, possibilitando um melhor detalhamento, no SPEM como componentes de processo e ciclo de vida no

CMMI demonstra que apesar de um grau de aderência significativo podem ganhar mais representatividade se seus resultados forem combinados.

Esta avaliação entre os modelos contribui para a concepção do meta-processo, fornecendo uma visão mais ampla de semelhanças e aspectos particulares de cada modelo.

## REFERÊNCIAS

- [1] CMMI Product Team (2002), Capability Maturity Model® Integration (CMMISM), Version 1.1.
- [2] Chrissis, M. B., Konrad, M., Shrum, S. (2003), CMMI®: Guidelines for Process Integration and Product Improvement, Addison Wesley.
- [3] Object Management Group. (2004), Software Process Engineering Metamodel Specification, Especificação adotada como padrão em janeiro de 2005.
- [4] Sampaio, A. T. F. (2004), “WEBPROCESS: UM PROCESSO ÁGIL PARA O DESENVOLVIMENTO DE APLICAÇÕES WEB”, Centro de Informática – Universidade Federal de Pernambuco.
- [5] Somerville, I. (6ª edição), “Engenharia de Software”, Addison Wesley.
- [6] Falbo, R A. (1998) “Integração de Conhecimento em um Ambiente de Desenvolvimento de Software”, Orientadora: Ana Regina Cavalcanti da Rocha.  
Tese de Doutorado, COPPE/UFRJ.
- [7] OLIVEIRA, Sandro; VASCONCELOS, Alexandre; ROUILLER, Ana Cristina. Uma Proposta de um Ambiente de Implementação de Processo de Software. Artigo a ser publicado na Revista InfoComp – Revista de Ciência da Computação vol. 4, n. 1, Lavras–MG, 2005.