

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
DEPARTAMENTO DE INFORMÁTICA  
MESTRADO EM INFORMÁTICA

**RODRIGO DAL MORO**

**Avaliação e Melhoria de Processos de Software:  
Conceituação e Definição de um Processo para  
Apoiar a sua Automatização**

Vitória  
2008

# **Livros Grátis**

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

Milhares de livros grátis para download.

**RODRIGO DAL MORO**

**Avaliação e Melhoria de Processos de Software:  
Conceituação e Definição de um Processo para  
Apoiar a sua Automatização**

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Orientador: Prof. Dr. Ricardo de Almeida Falbo.

**Vitória  
2008**

**RODRIGO DAL MORO**

**Avaliação e Melhoria de Processos de Software:  
Conceituação e Definição de um Processo para  
Apoiar a sua Automatização**

**Aprovada em 16 de julho de 2008.**

**COMISSÃO EXAMINADORA**

---

**Prof. Ricardo de Almeida Falbo, D.Sc.**

**Universidade Federal do Espírito Santo (UFES)**

**Orientador**

---

**Prof. Davidson Cury, D.Sc.**

**Universidade Federal do Espírito Santo (UFES)**

---

**Prof. Ana Regina Cavalcanti da Rocha, D.Sc.**

**Universidade Federal do Rio de Janeiro (COPPE-UFRJ)**

# **Dedicatória**

**Dedico esta conquista ao meu irmão e melhor amigo Maicon pelo apoio incondicional nos momentos mais difíceis desta jornada. Maicon, você sabe o quanto foi importante para que este momento acontecesse. Eu lhe disse que nunca iria esquecer o apoio que você me deu. Então, amigo, essa conquista é nossa.**

# Agradecimentos

Agradeço primeiramente a Deus por ter me dado forças, determinação e serenidade para superar todos os obstáculos na realização deste trabalho.

Agradeço também a todos, sem exceção, que torceram por mim, seja em orações, pensamentos positivos ou simplesmente com uma palavra amiga dada na hora certa. Pai, mãe, irmão, namorada, orientador, professores, amigos pessoais, de trabalho, do LabES, enfim, sei que cada um de vocês que lêem esses agradecimentos sabe exatamente onde se encaixar, pois seria injusto citar apenas alguns nomes. Saibam que vocês são pessoas maravilhosas e sem o apoio de todos eu não conseguiria alcançar mais esta vitória em minha vida. Desse modo, **MUITO OBRIGADO DE CORAÇÃO!**

Fugindo um pouco do padrão, gostaria de agradecer a mim mesmo pelo esforço, determinação e comprometimento que tive no decorrer desta jornada. Houve momentos difíceis, mas nessas horas pude contar com todos que torcem por mim. Então, levantei a cabeça e segui em frente. E, sinceramente, **VALEU A PENA!**

## SUMÁRIO

<b>Capítulo 1 – Introdução .....</b>	<b>01</b>
1.1 Motivação.....	01
1.2 Contexto e Objetivos.....	02
1.3 Histórico do Trabalho .....	04
1.4 Organização do Trabalho .....	06
 <b>Capítulo 2 – Avaliação e Melhoria de Processos de Software .....</b>	<b>07</b>
2.1 Introdução .....	07
2.2 Processo de Software .....	08
2.3 Avaliação e Melhoria de Processos de Software .....	12
2.3.1 Medição e Avaliação de Processos de Software.....	13
2.3.2 Melhoria de Processos de Software.....	16
2.4 Avaliação e Melhoria de Processos em Normas e Modelos de Qualidade de Processo de Software .....	19
2.4.1 Avaliação e Melhoria de Processos nas Normas ISO.....	19
2.4.2 Avaliação e Melhoria de Processos no CMMI.....	21
2.4.3 Avaliação e Melhoria de Processos no MPS.BR.....	22
2.5 Apoio Ferramental para a Avaliação e Melhoria de Processos de Software .....	25
2.5.1 Ferramenta APPRAISAL ASSISTANT.....	25
2.5.2 Ferramenta ProEvaluator .....	25
2.5.3 Ferramenta PISA .....	26
2.5.4 Ferramenta FAMP .....	26
2.5.5 Ferramenta AVALPRO .....	26
2.5.6 Ferramenta PILOT.....	27
2.5.7 Integração de Ferramentas de Automatização do Processo de Avaliação e Melhoria de Processos .....	27
2.6 Considerações Finais do Capítulo.....	28
 <b>Capítulo 3 – Um Processo de Avaliação e Melhoria de Processos de Software.....</b>	<b>29</b>
3.1 Introdução .....	29
3.2 Definição do Processo de AMP .....	30
3.3 Interação entre o Processo de AMP e Outros Processos.....	34

3.4 Funcionalidades para Apoiar o Processo de AMP Definido.....	37
3.4.1 Funcionalidades Gerais para Apoiar o Processo de AMP Definido.....	37
3.4.2 Funcionalidades Específicas para Apoiar o Processo de AMP Definido .....	38
3.5 Considerações Finais do Capítulo .....	41

<b>Capítulo 4 – Uma Ontologia para o Domínio de Qualidade de Software com foco em Avaliação e Melhoria de Processos de Software .....</b>	<b>42</b>
4.1 Introdução .....	42
4.2 Ontologias .....	44
4.3 O Método SABiO.....	46
4.4 A Ontologia de Qualidade de Software .....	48
4.4.1 A Sub-Ontologia de Modelo de Qualidade .....	49
4.4.1.1 Identificação de Propósito da Sub-Ontologia de Modelo de Qualidade .....	50
4.4.1.2 Captura e Formalização da Sub-Ontologia de Modelo de Qualidade .....	50
4.4.1.3 Avaliação da Sub-Ontologia de Modelo de Qualidade.....	52
4.4.1.4 Exemplos de Instanciação da Sub-Ontologia de Modelo de Qualidade .....	53
4.4.2 A Sub-Ontologia de Medição .....	54
4.4.2.1 Identificação de Propósito da Sub-Ontologia de Medição.....	54
4.4.2.2 Captura e Formalização da Sub-Ontologia de Medição .....	55
4.4.2.3 Avaliação da Sub-Ontologia de Medição .....	60
4.4.2.4 Exemplos de Instanciação da Sub-Ontologia de Medição .....	64
4.4.3 A Sub-Ontologia de Avaliação .....	66
4.4.3.1 Identificação de Propósito da Sub-Ontologia de Avaliação .....	66
4.4.3.2 Captura e Formalização da Sub-Ontologia de Avaliação .....	66
4.4.3.3 Avaliação da Sub-Ontologia de Avaliação .....	71
4.4.3.4 Exemplos de Instanciação da Sub-Ontologia de Avaliação.....	74
4.5 Trabalhos Correlatos na Literatura.....	76
4.6 Considerações Finais do Capítulo .....	78



<b>Capítulo 5 – Apoio Automatizado ao Processo de Avaliação e Melhoria de Processos de Software.....</b>	<b>79</b>
5.1 Introdução .....	79
5.2 O Ambiente de Desenvolvimento de Software ODE.....	80
5.3 AvaliaODE: Uma Ferramenta de Apoio à Avaliação de Processos .....	81
5.3.1 Modelo de Casos de Uso de AvaliaODE.....	82
5.3.2 Modelo Conceitual de AvaliaODE.....	84
5.3.3 Apresentação de AvaliaODE.....	88
5.4 MelhoriaODE: Uma Ferramenta de Apoio à Melhoria de Processos .....	94
5.4.1 Modelo de Casos de Uso de MelhoriaODE.....	95
5.4.2 Modelo Conceitual de MelhoriaODE.....	96
5.4.3 Apresentação de MelhoriaODE.....	99
5.5 Considerações Finais do Capítulo.....	104
<b>Capítulo 6 – Considerações Finais .....</b>	<b>105</b>
6.1 Conclusões .....	105
6.2 Perspectivas Futuras.....	110
<b>Referências Bibliográficas .....</b>	<b>113</b>

## LISTA DE TABELAS

Tabela 3.1 – Atividades do Processo AMP Definido.....	33
Tabela 3.2 – Identificação dos Processos do MPS.BR.....	35
Tabela 4.1 – Conceitos da Sub-Ontologia de Modelo de Qualidade .....	51
Tabela 4.2 – Avaliação da Sub-Ontologia de Modelo de Qualidade .....	53
Tabela 4.3 – Instanciação da Sub-Ontologia de Modelo de Qualidade .....	54
Tabela 4.4 – Conceitos da Sub-Ontologia de Medição .....	57
Tabela 4.5 – Conceitos Reutilizados da Ontologia de Processo de Software .....	58
Tabela 4.6 – Avaliação da Sub-Ontologia de Medição .....	61
Tabela 4.7 – Instanciação da Sub-Ontologia de Medição .....	64
Tabela 4.8 – Conceitos da Sub-Ontologia de Avaliação .....	68
Tabela 4.9 – Avaliação da Sub-Ontologia de Avaliação .....	72
Tabela 4.10 – Instanciação da Sub-Ontologia de Avaliação .....	75
Tabela 5.1 – Correlação entre os requisitos funcionais e os casos de uso de AvaliaODE.....	82
Tabela 5.2 – Correlação entre os requisitos funcionais e os casos de uso de MelhoriaODE....	94

## LISTA DE FIGURAS

Figura 2.1 – Abordagem em Níveis para a Definição de Processos.....	11
Figura 3.1 – Diagrama de Atividades do Processo AMP Definido.....	31
Figura 3.2 – Interação entre o processo AMP definido e demais processos MPS.BR.....	34
Figura 4.1 – Relacionamentos entre os tipos de ontologias (GUARINO, 1998) .....	45
Figura 4.2 – Perfil UML para representar ontologias e axiomas associados .....	47
Figura 4.3 – Dependência entre as ontologias de Qualidade de Software e de Processo em ODE.....	49
Figura 4.4 – O Modelo da Sub-Ontologia de Modelo de Qualidade.....	50
Figura 4.5 – O Modelo da Sub-Ontologia de Medição .....	56
Figura 4.6 – O Modelo da Sub-Ontologia de Avaliação .....	67
Figura 5.1 – Diagrama de Casos de Uso de AvaliaODE.....	83
Figura 5.2 – Diagrama de Pacotes de AvaliaODE .....	85
Figura 5.3 – Diagrama de Classes de AvaliaODE: Definição do Escopo de uma Avaliação...86	
Figura 5.4 – Diagrama de Classes de AvaliaODE: Resultados da Avaliação.....	88
Figura 5.5 – Tela principal de AvaliaODE.....	89
Figura 5.6 – Definição de Escopo de Avaliação em AvaliaODE – Dados Gerais .....	90
Figura 5.7 – Definição de Escopo de Avaliação em AvaliaODE – Seleção de Tipos de Artefatos .....	90
Figura 5.8 – Definição de Escopo de Avaliação em AvaliaODE – Seleção de Projetos .....	91
Figura 5.9 – Definição de Critérios de Avaliação em AvaliaODE .....	92
Figura 5.10 – Definição de Critérios de Avaliação em AvaliaODE .....	93
Figura 5.11 – Registro de Observação de Avaliação em AvaliaODE.....	93
Figura 5.12 – Registro de Parecer Final de Avaliação em AvaliaODE .....	94
Figura 5.13 – Diagrama de Casos de Uso de MelhoriaODE.....	95
Figura 5.14 – Diagrama de Pacotes de MelhoriaODE .....	96
Figura 5.15 – Diagrama de Classes de MelhoriaODE.....	98
Figura 5.16 – Tela principal de MelhoriaODE.....	99
Figura 5.17 – Inclusão de Sugestão de Melhoria em MelhoriaODE.....	100
Figura 5.18 – Inclusão de Sugestão de Alteração de Atividade em MelhoriaODE .....	100
Figura 5.19 – Dados Gerais de uma Proposta de Melhoria em MelhoriaODE .....	101
Figura 5.20 – Seleção de Sugestões de Melhoria para uma Proposta de Melhoria em MelhoriaODE .....	102

Figura 5.21 – Registro de Resultados para uma Proposta de Melhoria em MelhoriaODE.... 103

Figura 5.22 – Inclusão de um Resultado para uma Proposta de Melhoria em MelhoriaODE.104

## RESUMO

Dada a complexidade do processo de Avaliação e Melhoria de Processos de Software (AMP) e a sua forte inter-relação com outros processos do ciclo de vida de software, é preciso prover apoio automatizado para ele. Para ajudar a identificar requisitos funcionais, definiu-se um processo de AMP aderente ao MPS.BR, com o objetivo de guiar o estabelecimento de funcionalidades das ferramentas capazes de apoiar um processo de AMP de qualidade. Para estabelecer uma visão consensual sobre o domínio de AMP, a ser usada como modelo conceitual base, foi desenvolvida uma ontologia de qualidade de software com foco em AMP. Por fim, tomando por base a ontologia e o processo de AMP definidos, foram desenvolvidas duas ferramentas: AvaliaODE, responsável pelo apoio à avaliação de processos e produtos de software, e MelhoriaODE, responsável pelo apoio à melhoria de processos de software.

Este trabalho está inserido no contexto do Projeto ODE, um ambiente de desenvolvimento de software centrado em processo e baseado em ontologias, e tem por objetivo a definição de uma base conceitual para apoiar a automatização do processo de Avaliação e Melhoria de Processos de Software.

Palavras-chave: Processos de Software, Avaliação e Melhoria de Processos de Software, Ontologias, Ontologia de Qualidade de Software, Ambientes de Desenvolvimento de Software.

## **ABSTRACT**

Given the complexity of the Software Process Improvement and Assessment (SPIA) process and its strong inter-relationship with other processes in the software life cycle, we must provide automated support for it. To help identifying functional requirements, we define a SPIA process adherent to MPS.BR quality model, with the goal of guiding the establishment of tools functionalities that are able to support a quality SPIA process. To establish a consensual vision of the SPIA area to be used as a basic conceptual model, we developed an ontology of software quality with a focus on SPIA. Finally, based on the ontology and the SPIA process defined, we developed two tools: AvaliaODE, responsible for supporting software products and process evaluation, and MelhoriaODE, responsible for supporting software process improvement.

This work is inserted in the context of ODE Project, a process-centered software engineering environment that is developed based on ontologies. It aims to define a conceptual basis for automating the SPIA process.

**Keywords:** Software Process, Software Process Improvement and Assessment, Ontologies, Software Quality Ontology, Software Engineering Environments.

## Introdução

O objetivo deste capítulo é apresentar ao leitor as bases sobre as quais este trabalho foi construído, os objetivos buscados e a maneira como a pesquisa foi conduzida, procurando informar o leitor quanto à estrutura da dissertação e, de certa forma, motivá-lo para a leitura.

### *1.1 Motivação*

Organizações de software estão cada vez mais preocupadas com a qualidade dos produtos que desenvolvem. Essa preocupação se dá pelo fato delas estarem inseridas em um mercado bastante competitivo e globalizado, no qual o capital intelectual é o principal ativo considerado. Para se tornarem competitivas, portanto, as organizações não podem simplesmente se preocupar com o concorrente que está ao seu lado. A preocupação deve ir além das fronteiras, buscando sempre estar alinhada às expectativas e exigências relativas à qualidade de software.

Pensando nisso, muitas organizações estão despendendo esforços significativos na avaliação e melhoria de seus processos de software, buscando, com isso, a melhoria contínua dos mesmos, uma vez que estudos mostram que a qualidade do produto de software depende diretamente da qualidade dos processos adotados no seu desenvolvimento (FUGGETTA, 2000).

Dado o cenário acima, uma área de suma importância para que a organização alcance a melhoria contínua de seus processos é a Avaliação e Melhoria de Processos de Software. Uma vez que uma organização avalia constantemente seus processos, ela se conhece melhor e consegue analisar quais são os pontos críticos a serem melhorados. Além disso, esse autoconhecimento ajuda no estabelecimento e disseminação da cultura de processos dentro da organização (ZAHARAN, 1998), sendo fundamental para que haja apoio das pessoas envolvidas em iniciativas de melhoria de processos.

Entretanto, a área de Avaliação e Melhoria de Processos de Software é complexa e fortemente relacionada com outros processos do ciclo de vida de software, com destaque para a Medição. A medição deve ser ponto de atenção quando a organização busca a melhoria contínua de seus processos, sobretudo em níveis mais altos de

maturidade, pois é a base para a avaliação de um processo e, esta, por sua vez, é base para a melhoria de um processo.

Dada a complexidade do processo de Avaliação e Melhoria de Processos de Software e seus relacionamentos com outros processos, há a necessidade de se prover apoio automatizado para ele, o que requer, geralmente, diversas ferramentas. Neste contexto, os Ambientes de Desenvolvimento de Software (ADSs) se colocam como uma possibilidade de integração de ferramentas para apoiar de maneira consistente, além do processo de Avaliação e Melhoria de Processos de Software, os demais processos com os quais ele se relaciona, apoiando, assim, o processo de software como um todo, ou pelo menos porções significativas dele (HARRISON et al., 2000).

Entretanto, mesmo se tratando de um ambiente integrado, deve-se levar em conta que as ferramentas que o compõem dificilmente serão desenvolvidas pela mesma equipe e com os mesmos interessados e, portanto, é muito provável que algumas dessas ferramentas sejam construídas tomando por base diferentes conceituações. Assim, para que essas ferramentas funcionem adequadamente em conjunto, deve-se garantir que elas atribuam significados compatíveis para entidades do mundo real que fazem parte de seu domínio de interesse compartilhado.

Esse cenário foi a motivação principal para este trabalho, cujo foco é o apoio à automatização do processo de Avaliação e Melhoria de Processos de Software, baseado em uma ontologia para esse domínio.

## ***1.2 Contexto e Objetivos***

Este trabalho está inserido no contexto do Projeto ODE (*Ontology-based Software Development Environment*) (FALBO et al., 2003), um ADS centrado em processo e baseado em ontologias. Uma vez que em ODE as ferramentas são construídas baseadas em ontologias, a integração das mesmas é facilitada, pois os conceitos envolvidos são bem definidos e compartilhados, o que permite uma melhor interoperabilidade entre elas. Tal ambiente vem sendo desenvolvido no Laboratório de Engenharia de Software (LabES) da Universidade Federal do Espírito Santo (UFES).

Atualmente a base ontológica do ambiente ODE conta com diversas ontologias, dentre elas: Ontologia de Processo de Software (BERTOLLO, 2006), Ontologia de Gerência de Configuração de Software (NUNES, 2005) (ARANTES et al., 2007),



Ontologia de Requisitos de Software (NARDI, 2006), Ontologia de Organizações de Software (RUY, 2006) e Ontologia de Riscos de Software (FALBO et al., 2004).

Construídas a partir dessa base ontológica, há ferramentas para apoiar a definição de processos de software (BERTOLLO et al., 2006), acompanhamento de projetos (MORO et al., 2005), gerência de conhecimento (FALBO et al., 2004), gerência de riscos (FALBO et al., 2004a), estimativas (CARVALHO et al., 2006), gerência de configuração (NUNES et al., 2006), garantia da qualidade (PALACIO, 2007), engenharia de requisitos (MARTINS et al., 2006), dentre outras.

Entretanto ODE não conta com ferramentas específicas para apoiar o processo de Avaliação e Melhoria de Processos de Software (AMP). Desse modo, o objetivo geral deste trabalho é prover uma conceituação básica sobre esse domínio, a ser utilizada como base para o desenvolvimento de ferramentas de apoio ao processo de AMP em ODE.

Uma vez que em ODE as ferramentas são construídas com base em ontologias e com o intuito de se manter essa fundamentação teórica, há a necessidade de se buscar uma ontologia de qualidade de software que contemple o domínio de AMP. Mais ainda, dada a importância do processo de Medição para apoiar o processo de AMP, conceitos relativos à medição também devem ser considerados pela ontologia. No contexto do Projeto ODE já havia sido desenvolvida uma ontologia de qualidade de software (DUARTE e FALBO, 2000). Contudo, após realizar um estudo aprofundado da mesma, percebeu-se que ela não trata satisfatoriamente o domínio de AMP, uma vez que ela está fortemente baseada na avaliação da qualidade de produtos de software, tendo por base, sobretudo, a norma ISO/IEC 9126 (ISO/IEC 2001). Além disso, ela também trata de forma bem simplificada o processo de Medição, o que é considerado insuficiente para o apoio necessário ao processo de AMP. Assim, é um objetivo específico deste trabalho desenvolver uma ontologia de qualidade de software.

Outro objetivo específico deste trabalho é a definição de um processo de AMP, a partir do qual possam ser levantados requisitos funcionais para se prover apoio automatizado ao processo de AMP. Tal processo deve: (i) ser genérico, contemplando atividades e sub-atividades do processo; e (ii) ser baseado no modelo de qualidade MPS.BR (SOFTEX, 2007a), pois, com isso, há uma indicação que importantes boas práticas estão presentes no processo definido, uma vez que o MPS.BR é baseado nas principais normas e modelos internacionais de qualidade, a saber: CMMI (SEI, 2006a), ISO/IEC 12207 (ISO/IEC, 2008) e ISO/IEC 15504 (ISO/IEC, 2003).

Por fim, o último objetivo específico deste trabalho é desenvolver ferramentas de apoio à AMP, tomando por base o processo definido e a ontologia de qualidade de software proposta. Duas ferramentas com funcionalidades básicas foram desenvolvidas: AvaliaODE, para apoiar a avaliação de processos e produtos de software em ODE, e MelhoriaODE, visando prover algum apoio a iniciativas de melhoria de processos em ODE.

Dentre as funcionalidades de AvaliaODE podem-se citar: (i) definição do escopo de uma avaliação de processo, (ii) definição dos critérios de avaliação de processo, (iii) avaliação de processos e produtos do processo, dentre outras. Já em relação à ferramenta MelhoriaODE, suas funcionalidades incluem: (i) cadastro de sugestões de melhoria de processo, (ii) elaboração de propostas de melhoria de processo, e (iii) registro de resultados alcançados com a melhoria de um processo.

Assim, são objetivos específicos deste trabalho:

- Desenvolver uma Ontologia de Qualidade de Software, com foco no domínio de AMP e
- Definir um processo básico de AMP; e
- Desenvolver ferramentas para prover algum apoio automatizado ao processo de AMP definido, integrando-as ao ambiente ODE.

### ***1.3 Histórico do Trabalho***

Este trabalho teve início com uma revisão bibliográfica sobre a área de Avaliação e Melhoria de Processos de Software (AMP), na qual foram avaliados e discutidos artigos científicos, relatórios técnicos, livros, trabalhos acadêmicos e ferramentas de apoio ao processo de AMP. O foco desse estudo foi investigar a área de Avaliação e Melhoria de Processos de Software para entender seus conceitos e levantar requisitos a serem tratados no apoio à automatização do processo de AMP, o foco deste trabalho.

Percebeu-se com esse estudo inicial que a área de Medição está fortemente relacionada com a área de Avaliação e Melhoria de Processos de Software, uma vez que a medição é base para a avaliação de um processo, sobretudo em níveis mais altos de maturidade. Desse modo, um estudo sobre a área de Medição também foi realizado.

Além disso, durante esse estudo fez-se o esboço inicial de um processo de AMP genérico e bem fundamentado para utilizá-lo como base para o estabelecimento de

requisitos funcionais para as ferramentas de apoio à AMP. Para tal foram estudadas diversas normas e modelos de qualidade de processo de software para fundamentar a definição do processo de AMP. A partir desse estudo, decidiu-se basear o processo de AMP no modelo de qualidade MPS.BR e, além disso, entender seus relacionamentos com os demais processos do MPS.BR, mostrando com isso, a complexidade e amplitude do domínio do processo de AMP.

A partir desses estudos, mais especificamente das áreas de Avaliação e Melhoria de Processos de Software e Medição, decidiu-se analisar a Ontologia de Qualidade de Software desenvolvida por DUARTE e FALBO (2000), parte constituinte da base ontológica de ODE. Essa análise mostrou que essa ontologia estava fortemente baseada na avaliação de produtos de software e que tratava de forma simplificada o processo de medição. Desse modo, optou-se pelo desenvolvimento de uma nova ontologia. Para tal, além da análise da ontologia de qualidade proposta em (DUARTE e FALBO, 2000), outros trabalhos na área também foram analisados, dentre eles (MARTIN e OLSINA, 2003) e (BERTOA et al., 2006).

Uma vez definidos o processo de AMP e a nova ontologia de qualidade de software, foram discutidas as funcionalidades necessárias para apoiar de forma automatizada o processo de AMP proposto. Com base nas funcionalidades identificadas, definiu-se o escopo das ferramentas a serem desenvolvidas neste trabalho, tendo sido desenvolvidas duas ferramentas: AvaliaODE e MelhoriaODE. O objetivo principal de AvaliaODE é apoiar a avaliação de processos de software. Já MelhoriaODE tem por objetivo principal o apoio à melhoria de processos de software. Vale ressaltar que as duas ferramentas foram desenvolvidas segundo os conceitos definidos pela Ontologia de Qualidade de Software e estão integradas ao ambiente ODE. Outro ponto importante a ser ressaltado foi a reestruturação da ferramenta GQA-ODE (PALACIO, 2007), a ferramenta de garantia de qualidade de ODE, para deixá-la aderente à ontologia desenvolvida. Essa reestruturação foi necessária porque funcionalidades de GQA-ODE foram compartilhadas com as ferramentas AvaliaODE e MelhoriaODE.

Por fim, é importante citar que as ferramentas desenvolvidas, AvaliaODE e MelhoriaODE, junto com GQA-ODE, cobrem apenas parte das funcionalidades necessárias para apoiar um processo de AMP.

## 1.4 Organização do Trabalho

Esta dissertação contém, além deste capítulo, mais cinco capítulos, a saber:

- *Capítulo 2 – Avaliação e Melhoria de Processos de Software*: fornece uma fundamentação teórica sobre Avaliação e Melhoria de Processos de Software no contexto da busca pela melhoria contínua dos processos organizacionais.
- *Capítulo 3 – Um Processo de Avaliação e Melhoria de Processos de Software*: apresenta o processo básico de Avaliação e Melhoria de Processo de Software (AMP) definido tomando por base o modelo de qualidade MPS.BR. Apresenta, também, a interação entre o processo AMP e outros processos do MPS.BR que têm afinidade com o processo AMP. Por fim, são discutidas funcionalidades requeridas para prover apoio automatizado à AMP, com base no processo proposto.
- *Capítulo 4 – Uma Ontologia para o Domínio de Qualidade de Software com foco em Avaliação e Melhoria de Processos de Software*: apresenta a Ontologia de Qualidade de Software desenvolvida seguindo a abordagem proposta pelo método SABiO e mostrando, também, sua instanciación para as perspectivas de processo e produto de software.
- *Capítulo 5 – Apoio Automatizado do Processo de Avaliação e Melhoria de Processos de Software*: apresenta as ferramentas AvaliaODE e MelhoriaODE, desenvolvidas para apoiar parcialmente o processo de AMP proposto. Como as duas ferramentas foram desenvolvidas no contexto do ambiente ODE, o mesmo é sucintamente apresentado.
- *Capítulo 6 – Considerações Finais*: apresenta as conclusões sobre o trabalho desenvolvido e apresenta as contribuições e as perspectivas para trabalhos futuros.

### **Avaliação e Melhoria de Processos de Software**

O objetivo deste capítulo é estabelecer uma fundamentação teórica sobre Avaliação e Melhoria de Processos de Software no contexto da busca pela melhoria contínua dos processos organizacionais. Desse modo, a abordagem adotada para este capítulo se dá por: (i) uma apresentação do conceito de processo de software; (ii) uma discussão sobre avaliação e melhoria de processos de software em organizações; (iii) uma apresentação da visão de avaliação e melhoria de processos de alguns dos principais modelos e normas de qualidade de software; e, por fim, (iv) um levantamento de ferramentas existentes de apoio aos processos de avaliação e melhoria de processos de software organizacionais.

#### ***2.1 Introdução***

Um dos principais motivos para que organizações de software adotem uma visão de melhoria contínua de seus processos de software, ou seja, se preocupem constantemente com a avaliação e melhoria de seus processos, é o fato da qualidade do produto final depender diretamente da qualidade do processo de desenvolvimento adotado. Assim, quanto maior a preocupação e o esforço para que seus processos estejam alinhados aos objetivos de negócio, maiores serão os benefícios alcançados pela organização. Além disso, a melhoria contínua dos processos faz com que a organização se posicione de uma forma bastante competitiva no mercado em que atua.

Contudo, antes de uma organização de software pensar em melhoria, é importante que ela, primeiramente, tenha maturidade para definir seus processos. Essa maturidade não é algo trivial de ser alcançado, pois passa tanto pelo apoio da alta gerência da organização em relação a investimentos em qualidade, quanto pela cultura organizacional, em que todos, nos vários níveis hierárquicos, devem entender os reais benefícios de se utilizar processos em seu cotidiano.

Uma vez definidos os processos organizacionais, outro ponto importante e fundamental para que ocorram melhorias é a existência de um processo de avaliação de processos. Neste ponto, as organizações devem se preocupar em avaliar seus processos periodicamente com o intuito de melhorá-los ou otimizá-los a partir da análise dos resultados das avaliações. Pode-se perceber, então, que a melhoria de processos ocorre dentro do contexto de uma organização, sendo utilizada como base para tratar os seus

objetivos. Logo, é de responsabilidade da alta gerência encorajar a participação das pessoas que irão executar o processo nas atividades de melhoria e, além disso, facilitar e gerenciar as atividades de melhoria de processos inerentes à sua organização (CHRISSIS et al., 2003).

Para ZAHRAN (1998) a melhoria da qualidade, seja em processos de software, seja em qualquer outra área, está intimamente ligada à necessidade de adaptação da organização, com uma mudança de ênfase em produto para uma organização focada em processos. Essa mudança de foco proporciona, entre outros, os seguintes benefícios: (i) alinhamento entre as atividades e o foco estratégico da organização; (ii) consistência entre as várias atividades desempenhadas por pessoas diferentes, mas com objetivos comuns; e (iii) possibilidade de medição e maior facilidade na condução das equipes.

Assim, é essencial que a organização entenda que a melhoria de processos de software é muito importante, pois vai auxiliá-la na diminuição de redundâncias, aumento da produtividade e aumento da capacidade gerencial de seus processos organizacionais (UMARGI et al., 2005).

Como pode ser observado pelas ponderações acima, avaliar e melhorar processos tornam-se metas fundamentais para que uma organização de software possa, além de aumentar sua maturidade no processo de software, estar alinhada às tendências do mercado no qual atua. A procura pela melhoria contínua de processos tende a deixá-la mais preparada para vencer desafios e mudanças constantes que o mercado globalizado e competitivo de software apresenta.

Este capítulo está estruturado da seguinte forma: a Seção 2.2 fornece uma base teórica acerca de processo de software; a Seção 2.3 fornece uma visão geral sobre avaliação e melhoria de processos de software; na Seção 2.4 são sucintamente apresentados os principais modelos e normas de qualidade de software, procurando destacar como eles tratam a avaliação e a melhoria de processos de software; na Seção 2.5 são discutidas questões relativas ao apoio automatizado à avaliação e melhoria de processos de software; por fim, a Seção 2.6 apresenta as considerações finais do capítulo.

## ***2.2 Processo de Software***

Ao se pensar em desenvolver um software, sempre vêm à mente questões referentes a como será esse desenvolvimento, tais como: (i) Que tipo de modelo de ciclo

de vida será utilizado? (ii) Quais as atividades e em que ordem elas devem ser executadas? (iii) Quais são os produtos de trabalho gerados e consumidos em cada uma das atividades? (iv) Quais recursos serão utilizados e em quais atividades do processo eles serão necessários? (v) Quais procedimentos devem ser utilizados na realização das diversas atividades? Enfim, para se ter uma boa resposta para todas essas questões, é fundamental pensar primeiro em definir o processo a ser utilizado no desenvolvimento desse software. Uma vez definido o processo de software a ser seguido, a probabilidade de sucesso do projeto em termos de qualidade do produto final aumenta, visto que já se sabe, a priori, o que deve ser feito, por quem e em que momento do desenvolvimento do software.

Mas afinal, o que é um processo de software? Um processo de software pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software. Sendo assim, um processo de software define o conjunto de atividades que serão conduzidas no contexto do projeto, os recursos (*software*, *hardware* e pessoas) necessários, os artefatos (insumos e produtos) e os procedimentos a serem adotados na realização de cada uma das atividades (GRUHN, 2002) (MORO et al., 2005).

O desenvolvimento de produtos de software confiáveis e dentro do cronograma e custo planejados sempre foi um desafio para a Engenharia de Software (BERTOLLO et al., 2003). Estudos apontam que uma das principais causas dos problemas enfrentados no desenvolvimento de software é a falta de um processo bem definido e efetivo (FUGGETTA, 2000). Assim, as organizações de software têm dado uma maior ênfase na utilização de processos como uma forma de buscar, principalmente, maior qualidade e controle sobre custos e prazos em projetos de software (ANDRADE et al., 2004).

Um processo de software definido permite monitorar o tempo gasto em cada tarefa e o número de defeitos introduzidos ou removidos em cada etapa do desenvolvimento. Esses dados auxiliam a análise do processo e a identificação de problemas e possíveis melhorias (ZAHARAN, 1998).

Pode-se notar que, além de apoiar o desenvolvimento do software com qualidade, ter um processo definido abre espaço também para melhorias contínuas dentro da organização, isto é, apóia a organização a estar sempre evoluindo em termos da qualidade de seus produtos. Logo, é imprescindível que uma organização, preocupada com a qualidade de seus produtos, esteja preocupada em primeiro lugar em definir seus processos de software.

De fato, a definição de processos de software é fator fundamental para se aumentar a probabilidade de sucesso de um projeto de software. Mas é claro que apenas definir processos não é suficiente. É importante que o processo seja executado conforme o planejado e monitorado em sua execução para que os reais benefícios sejam alcançados ao final do projeto.

Além disso, dada a complexidade da definição de processos, uma estratégia advogada pela maioria das normas e modelos de qualidade de processo consiste em definir processos padrão organizacionais. Um processo padrão descreve as atividades que devem ser realizadas em todos os projetos da organização e os demais ativos de processo envolvidos (artefatos, procedimentos, recursos etc). Assim, usando um processo padrão como base para o planejamento do processo de software específico do projeto, gerentes de projeto têm sua tarefa facilitada, além de ser mais direta a definição de planos em conformidade com os padrões de qualidade da organização (BERGER, 2003).

No entanto, um processo padrão não atende necessariamente a todos os tipos de projetos de uma organização. Como o próprio nome sugere, ele é um padrão organizacional que deve ser seguido em todos os projetos da organização. Contudo, para ser abrangente, ele normalmente está definido num alto nível de abstração e, sendo assim, pode e deve ser modificado, quando necessário, acrescentando-se características específicas para um projeto de um determinado tipo que exija particularidades.

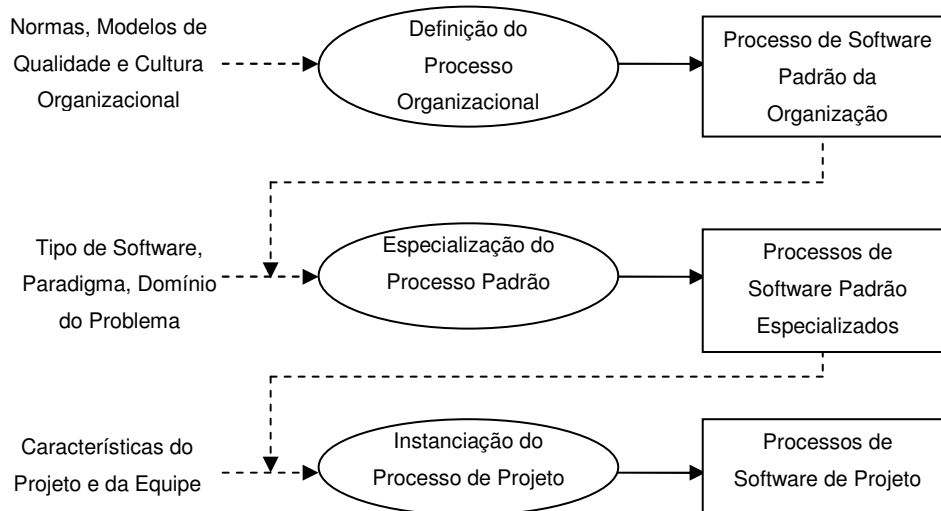
Uma forma de apoiar a instanciação do processo padrão para tipos de projetos com características diferentes é por meio de uma abordagem de definição de processos em níveis (ROCHA et al., 2001). Nessa abordagem, como pode ser observado pela figura 2.1, são definidos três níveis de processos, a saber: (i) definição do processo organizacional, (ii) especialização do processo padrão e (iii) instanciação do processo de projeto.

A definição do processo organizacional tem por objetivo definir o processo de software padrão da organização, contemplando apenas os ativos de processo essenciais que devem ser incorporados a quaisquer processos da organização. Idealmente, o processo padrão da organização deve ser definido considerando padrões e modelos de qualidade, como CMMI, ISO/IEC 12207 e MPS.BR. Além disso, a cultura organizacional e a prática da organização em engenharia de software influenciam diretamente na definição do seu processo de software padrão.



O segundo nível é responsável pela definição das características específicas de um determinado tipo de projeto. Nesse nível são desenvolvidos processos padrão especializados para os tipos de projetos mais comuns dentro da organização. Características como o tipo de software a ser desenvolvido, o paradigma utilizado e o domínio de aplicação do software podem ser consideradas para essa especialização. Durante a especialização, ativos de processo podem ser adicionados ou modificados de acordo com o contexto da especialização.

Por fim, o terceiro nível da definição de processos diz respeito à instanciação do processo de um projeto. Nesse momento, o processo padrão especializado mais indicado pode ser instanciado para um projeto específico. Durante a instanciação do processo de projeto, particularidades desse projeto e características da equipe devem ser levados em conta. Além disso, o modelo de ciclo de vida deve ser definido e novas atividades, assim como artefatos consumidos e produzidos, recursos requeridos e procedimentos, podem ser adicionados.



**Figura 2.1 – Abordagem em Níveis para a Definição de Processos.**

No entanto, definir processos padrão e especializados não é suficiente. Assim como as próprias organizações, os processos não são estáticos e necessitam acompanhar a dinâmica das organizações e de seus negócios. A definição de processos organizacionais é apenas o ponto de partida. Medir, avaliar e melhorar continuamente os processos são atividades igualmente importantes. Assim, um processo de Avaliação e Melhoria de Processos (AMP) deve ser estabelecido.

### ***2.3 Avaliação e Melhoria de Processos de Software***

Organizações de software têm demonstrado muito interesse pela melhoria contínua de seus processos. Elas perceberam que a melhoria contínua de processos está diretamente ligada ao nível de maturidade em desenvolvimento de produtos de software em que elas se encontram. Perceberam, ainda, que podem alinhar a melhoria de seus processos organizacionais aos seus objetivos estratégicos de negócio, tornando-se, assim, organizações mais atentas ao mercado em que atuam e focadas na busca pela excelência na qualidade de software (PFLEEGER, 2004).

Sob esta ótica, organizações que realizam a melhoria contínua de seus processos tendem a ter uma maior consciência do seu potencial e de suas principais características positivas, além é claro, de seus pontos fracos e oportunidades de melhoria. Essa consciência organizacional ajuda no estabelecimento e divulgação de uma cultura de processos dentro da organização (ZAHARAN, 1998), sendo fundamental para que haja apoio das pessoas envolvidas em iniciativas de melhoria de processos.

A melhoria de um processo é, sobretudo, um processo de mudança (ZAHARAN, 1998) e, sendo assim, necessita ser analisada sob o ponto de vista de todos os envolvidos nessa mudança. A tendência natural das pessoas é ser resistente às mudanças. Então, para se conseguir êxito em uma melhoria de processos, é preciso observar os pontos de vista dos diferentes envolvidos. A alta gerência está normalmente preocupada com resultados e custos. Por outro lado, desenvolvedores estão mais preocupados com o trabalho realizado em suas atividades cotidianas (CONRADI et al., 2002). Assim, é muito importante que se consiga uma visão de melhoria de processos em que ambas as partes entendam a importância da melhoria.

A melhoria de processos envolve, em muitos casos, o estudo e a utilização de novas ferramentas e técnicas de coleta e análise de dados. Essas iniciativas podem causar desconforto e desconfiança nas pessoas, caso a organização e sua equipe de processos não deixem claros os objetivos das melhorias (UMARGI et al., 2005).

Tendo em vista o exposto, definir e divulgar aos envolvidos os objetivos que devem ser atingidos com uma melhoria de processo são ações essenciais para o sucesso dessa empreitada. Mais ainda, para que os objetivos determinados para uma melhoria sejam atingidos, é fundamental que seja estabelecida uma gestão adequada dos processos envolvidos na melhoria contínua (CAMPOS, 2005). Neste contexto, uma abordagem interessante é considerar uma melhoria de processo como um projeto

organizacional, de modo que as práticas de planejamento, execução, monitoramento e controle de um projeto sejam aplicadas, aumentando, com isso, a probabilidade de sucesso do projeto de melhoria.

Contudo, conforme discutido na seção anterior, antes de pensar em melhoria de processos é importante que a organização se preocupe em definir seus processos organizacionais. É fundamental que uma organização tenha seus processos implantados, pois, sem isso, não é possível avaliá-los nem melhorá-los (WANG, 2002). Em suma, definir processos é fundamental, mas insuficiente. Processos não podem ser definidos e mantidos “congelados” para sempre. Processos precisam continuamente passar por mudanças e refinamentos para aumentar a sua habilidade de lidar com requisitos e expectativas da organização e do mercado no qual ela atua. Assim, processos precisam ser continuamente avaliados e melhorados (FUGGETTA, 2000).

A avaliação sistemática da qualidade de um processo, de seus ativos e de seus produtos resultantes é essencial para apoiar a implementação de estratégias de melhoria dentro de uma organização (FUGGETTA, 2000). Contudo, uma avaliação sistemática não pode ser conduzida em bases meramente subjetivas. Assim, é necessário identificar características capazes de indicar a qualidade de um processo, medir essas características, analisar os resultados dessas medições e concluir sobre as necessidades de melhoria. Assim, a melhoria contínua passa pela medição e avaliação de processos para, aí sim, mediante informações objetivas, a organização poder tomar a decisão sobre a necessidade e a viabilidade de uma determinada melhoria. Portanto, para se falar de melhoria é necessário falar também de medição e avaliação.

### **2.3.1 Medição e Avaliação de Processos de Software**

Organizações de software, de modo geral, estão sempre em busca de responder questões como: (i) Estamos alcançando os resultados que desejamos? (ii) Nossos clientes estão satisfeitos com nossos produtos e serviços? (iii) Podemos reduzir nossos custos aumentando nossa produtividade? Essas questões não são fáceis de serem respondidas pelo simples fato de que, muitas vezes, não se sabe exatamente o que é desejado, o que deixa o cliente satisfeito e nem mesmo qual é a produtividade atual.

Para responder essas questões, é fundamental que a organização tenha informações objetivas acerca de seu desempenho. Tomando o exemplo da produtividade, a organização precisa, antes de tudo, conhecer sua atual produtividade para aí sim, poder pensar em aumentá-la. Ou seja, é preciso implantar um processo de

medição. O processo de medição tem um papel fundamental na melhoria de processos, uma vez que, além de propiciar à organização um autoconhecimento, também apóia a gestão dos processos e a identificação de oportunidades de melhorias (FLORAC e CARLETON, 2000).

Outro ponto importante referente ao processo de medição é o fato da medição prover informações objetivas que ajudam uma organização a tomar decisões avaliando o impacto, positivo ou negativo, em seus negócios e no desempenho de seus projetos. Um processo de medição torna-se mais eficiente quando implementado levando-se em consideração os objetivos de negócio da organização e, além disso, quando integrado com as atividades técnicas e gerenciais de um projeto de software (McGARRY et al., 2002).

Segundo a norma ISO/IEC 15939 (ISO/IEC, 2002), o objetivo da medição em software é coletar, analisar e reportar dados relacionados com os produtos desenvolvidos e processos implementados na organização e em seus projetos, para apoiar a gestão efetiva dos processos e demonstrar objetivamente a qualidade dos produtos.

No entanto, apesar de seus benefícios evidentes, o processo de medição não é de fácil implementação. É preciso que a organização tenha maturidade em saber o que deve, mas, principalmente, o que não deve medir, pois caso contrário, terá uma grande quantidade de dados sem conseguir transformá-los em informações úteis para a organização.

Pensando nisso, PFLEEGER (1997) definiu um conjunto de orientações para que as medições sejam realmente efetivas dentro das organizações, como por exemplo: (i) usar estratégias diferentes para culturas diferentes; (ii) não fazer medições no escuro; (iii) ser criterioso no processo de medição; (iv) ser simples e direto; (v) mudar as medições e seus modelos à medida que se entenda mais a situação; (vi) utilizar alguma medida é melhor do que não utilizar nenhuma; (vii) medir duas vezes, agir uma; (viii) deixar as medidas próximas aos desenvolvedores; (ix) capturar o máximo possível sem incomodar os desenvolvedores e (x) iniciar o processo de medição com quem está precisando de ajuda.

Já KAN (2003) definiu duas características fundamentais para o processo de medição: confiabilidade e validade. A confiabilidade refere-se à consistência dos resultados das medições realizadas, ou seja, utilizando o mesmo método de medição no mesmo objeto, devem-se obter os mesmos resultados. Já a validade de uma medição diz

respeito a quanto uma medição ou uma medida realmente mede o que se pretende medir. Quanto mais abstrato for o objeto de medição, maior a probabilidade de ocorrerem problemas de confiabilidade e validade nas medições.

Para tentar minimizar a dificuldade de implantação do processo de medição, deve-se elaborar um plano de medição. De um modo geral, para gerar um plano de medição, é necessário, primeiro, definir que entidades (processos, produtos, recursos) serão objeto de medição e quais de suas características serão medidas. Segundo, é preciso definir quais medidas serão usadas para quantificar essas características. Medidas podem ser básicas, i.e. funcionalmente independentes de outras medidas, ou derivadas, definidas como uma função de outras medidas (ISO/IEC, 2002). Uma vez planejada a medição, pode-se passar efetivamente à sua execução, o que envolve a aplicação de procedimentos de medição para se obter um valor para uma medida da entidade em questão.

Uma vez realizada a medição, devem-se analisar seus resultados para avaliar a qualidade das entidades medidas, gerando observações tais como pontos fortes, pontos fracos, não conformidades e oportunidades de melhoria, que servirão de base para a melhoria do processo que está sendo avaliado. É importante verificar quais são os pontos que precisam de maior atenção da organização, pois assim ela poderá priorizar os processos que necessitam de uma melhoria em caráter mais emergencial (SOMMERVILLE e RANSOM, 2005). Além disso, essas observações também podem dar origem a ações de qualidade, que podem ser tanto ações de melhoria, como também ações preventivas ou corretivas.

Os objetivos da avaliação de processos são: (i) identificar as áreas prioritárias para melhorias e prover algum tipo de orientação sobre como fazer essas melhorias, (ii) determinar com que extensão os processos da organização contribuem para se atingir os objetivos de negócio e (iii) apoiar a organização na realização das melhorias contínuas identificadas (ISO/IEC, 2008) (HUMPHREY, 1989).

A avaliação de processos é, na maioria dos casos, parte de um programa de abrangência maior, como, por exemplo, uma iniciativa de melhoria de processos de software ou uma avaliação de capacidade de software. Ou seja, a avaliação não é um fim, mas sim o catalisador que inicia o processo de mudança dentro de uma organização (ZAHARAN, 1998). Com a realização da medição e, em seguida, da avaliação de um processo, há informações suficientes para a organização tomar uma decisão pela melhoria de um processo. Desse modo, juntas, medição e avaliação têm um papel

fundamental para se conhecer a situação atual dos processos e, portanto, são a base para a sua melhoria.

### **2.3.2 Melhoria de Processos de Software**

As organizações de software que buscam uma melhoria contínua em seus processos são aquelas que, via de regra, estão entre as mais bem sucedidas no mercado em que atuam.

Uma organização preocupada em melhorar seus processos continuamente deve entender, antes de tudo, que a melhoria de processos não é um fim, mas sim um meio utilizado para aperfeiçoar seus processos organizacionais, visando, com isso, a melhoria na qualidade de seus produtos (HARDGRAVE et al., 2005).

WANG (2002) define um modelo de melhoria de processos como sendo um modelo operacional que provê orientação para a melhoria da capacidade de um sistema de processos pela mudança, atualização ou aperfeiçoamento dos processos existentes, com base nos resultados de uma avaliação.

Já para o SEI (2006a), programas de melhoria de processos têm por objetivo operacionalizar a institucionalização de modelos e normas em uma organização, sendo tratados como projetos e geralmente suportados por uma visão estratégica da organização.

Seja qual for a abordagem adotada, uma organização ao iniciar a melhoria dos seus processos busca, dentre outros (FLORAC e CARLETON, 2000): (i) entender as características dos processos existentes e as questões que afetam a sua capacidade; (ii) planejar, justificar e implementar ações que modificarão os processos, tornando-os mais coerentes com as necessidades do negócio e (iii) avaliar os impactos e benefícios resultantes e compará-los com os custos advindos das mudanças realizadas.

É importante ressaltar que a melhoria de processos é um processo amplo e complexo que envolve todos os níveis hierárquicos de uma organização, necessitando do apoio de todos para alcançar seus objetivos. Logo, definir em âmbito organizacional quais são os objetivos esperados para uma melhoria de processos é fator fundamental para conseguir êxito nessa tarefa.

Para que uma organização atinja os objetivos esperados com uma melhoria de processos, uma abordagem interessante é estabelecer um Programa de Melhoria de Processos, que envolva, além da melhoria em si, também a medição e a avaliação de seus processos organizacionais. VARKOI (2002) estruturou um Programa de Melhoria

de Processos em três grandes fases, a saber: (i) análise das necessidades, apoiada pelo processo de medição, quando são apresentados os processos atuais e a identificação e priorização das necessidades de melhoria desses processos; (ii) avaliação dos processos, apoiada pelo processo de avaliação, quando os processos são avaliados de forma planejada; e (iii) ações de melhoria, apoiada pelo processo de melhoria, quando as ações de melhoria são planejadas, executadas e implementadas na organização.

Entretanto, para que um Programa de Melhoria de Processos obtenha o sucesso esperado, este deve satisfazer a requisitos estratégicos, culturais e técnicos (CAMPOS, 2005). É importante que todas as visões sejam atendidas promovendo um equilíbrio entre as partes envolvidas na melhoria de processos.

Acima de tudo, uma melhoria de processo deve estar alinhada aos objetivos estratégicos da organização. De nada adianta melhorar processos se a melhoria realizada estiver fora do foco organizacional. Ao contrário, uma vez que a alta direção da organização perceba que os esforços estão sendo dispendidos em melhorias desalinhadas aos seus objetivos estratégicos, corre-se um alto risco do Programa de Melhoria de Processos ser desacreditado, parcialmente desmobilizado ou definitivamente cancelado. Logo, alinhar a melhoria de processos aos objetivos estratégicos e organizacionais é essencial para o sucesso do Programa de Melhoria de Processos dentro de uma organização.

Contudo, é importante lembrar, também, que uma organização é composta por pessoas e, além disso, existe uma cultura organizacional previamente estabelecida. Desse modo, conforme mencionado anteriormente, é importante que a cultura organizacional seja considerada, pois dependendo do tipo de cultura e das crenças e experiências dos envolvidos, algumas melhorias podem não obter os resultados esperados (ALBUQUERQUE, 2005).

Por fim, a institucionalização da melhoria de processos dentro de uma organização requer o apoio de uma infra-estrutura de processo, que sirva de suporte para as atividades de melhoria ZAHARAN (1998). Há basicamente dois aspectos relativos à questão de infra-estrutura que devem ser levados em conta, a saber ZAHARAN (1998): (i) infra-estrutura organizacional, ou seja, relativa à gerência e suas atividades e (ii) infra-estrutura tecnológica, ou seja, relativa às ferramentas de apoio ao processo de melhoria de processos. Também é considerada uma questão de ordem técnica a adoção de normas e modelos de qualidade de software.

Os aspectos técnicos podem ser mais facilmente atingidos e observados, uma vez que se adote uma norma ou modelo de referência. Porém, essa facilidade pode se tornar um problema, na medida em que muitos programas de melhoria têm dado demasiada atenção a aspectos técnicos dos modelos e deixando de lado fatores de negócio e organizacionais (CATTANEO et al., 2001). Isto acaba levando a situações de organizações que têm elevados níveis de avaliação nos modelos de maturidade, mas não conseguem melhorias efetivas em aspectos como: redução de tempo de projeto, redução de custos, aumento da satisfação dos clientes, aumento da participação no mercado, dentre outros.

A melhoria de processos de software, em linhas gerais, envolve (ALBUQUERQUE, 2005): (i) identificar pontos fortes e fracos, (ii) implementar as devidas alterações, (iii) avaliar os impactos e benefícios decorrentes das modificações realizadas e (iv) institucionalizar na organização as melhorias identificadas como benéficas. Um processo pode ser melhorado por meio da introdução, exclusão ou alteração de atividades no processo (AHONEN et al., 2002), o que pode envolver a exclusão, alteração ou introdução de práticas, procedimentos, ferramentas, orientações etc.

Uma vez bem planejada e conduzida, a melhoria de processos de software pode gerar bons resultados para a organização. Contudo, como todo projeto, a melhoria de um processo também enfrenta problemas e dificuldades que devem ser considerados, dentre eles: (i) as empresas necessitam enxergar retornos reais para continuarem com seu Programa de Melhoria de Processos (CONRADI e FUGGETTA, 2002); (ii) apesar da formação profissional ser um componente central da melhoria de processos, poucas organizações atualizam com regularidade a capacidade da sua força de trabalho em processos e métodos. Assim, é comum a melhoria de processos não andar em paralelo com atividades de melhoria dos recursos humanos (MASON, 2003); (iii) programas de melhoria que não são implantados logo após a avaliação dos processos tendem a apresentar mais dificuldades, pois, com o tempo, a expectativa da equipe diminui, bem como o grau de clareza e compreensão dos problemas (MEHNER, 1999).

Percebe-se que a melhoria de processos não é um processo trivial de ser implementado em uma organização. Portanto, prover apoio automatizado, preferencialmente baseado em normas e modelos de qualidade, a esse processo é muito importante.



## ***2.4 Avaliação e Melhoria de Processos em Normas e Modelos de Qualidade de Processo de Software***

A importância da avaliação e melhoria de processos é reconhecida por vários modelos e normas relacionados à qualidade de processo de software. Nesta seção apresentamos sucintamente os processos relacionados, bem como os métodos de avaliação, quando pertinente.

### **2.4.1 - Avaliação e Melhoria de Processos nas Normas ISO**

As normas internacionais ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software (ISO/IEC, 2008) e ISO/IEC 15504 – Tecnologia da Informação – Avaliação de Processos (ISO/IEC, 2003) estabelecem um *framework* com terminologias bem definidas e boas práticas para a definição e avaliação de processos de software.

Pensando em avaliação e melhoria de processos, elas fazem um par consistente, uma vez que a ISO/IEC 12207 define os processos para melhoria de processos e a ISO/IEC 15504 foca em processos referentes à avaliação de processos. Mais ainda, para confirmar essa consistência, a ISO/IEC 12207 é utilizada como um exemplo de modelo de referência para as avaliações da ISO/IEC 15504, na parte 5 desta última.

A norma ISO/IEC 12207 estabelece uma estrutura comum para os processos de ciclo de vida de software. Sua estrutura contém processos, atividades e tarefas que devem ser aplicados na aquisição de sistemas e produtos e serviços de software, e no fornecimento, desenvolvimento, operação, manutenção e distribuição de produtos de software ou porções de software de um sistema, realizados interna ou externamente à organização. Esse conjunto de processos, atividades e tarefas foi projetado para ser adaptado, de acordo com as características específicas, a uma organização, projeto ou aplicação, o que pode envolver o detalhamento, a adição e a supressão de elementos de acordo com o objetivo de utilização.

Essa norma define um Processo de Gerência de Modelo de Ciclo de Vida, cujo objetivo é definir, manter e assegurar a disponibilidade de políticas, processos de ciclo de vida, modelos de ciclo de vida e procedimentos para uso pela organização. Esse processo provê políticas, processos e procedimentos de ciclo de vida que são consistentes com os objetivos da organização, que são definidos, adaptados, melhorados e mantidos para apoiar necessidades de projetos específicos dentro do contexto da

organização, e que são passíveis de serem aplicados usando métodos e ferramentas efetivos e experimentados. Ele envolve três macro-atividades:

- Estabelecimento do Processo: visa estabelecer um conjunto de processos organizacionais e modelos de ciclo de vida que se aplica às suas atividades de negócio;
- Avaliação do Processo: tem como objetivo determinar como a organização deve desenvolver, documentar e aplicar os procedimentos de avaliação de processos. Além disso, também apóia a realização do planejamento e revisão dos processos em períodos pré-determinados, garantindo, com isso, a eficiência de seus processos sob a ótica dos resultados práticos de suas avaliações;
- Melhoria do Processo: visa melhorar os processos da organização baseando-se nos resultados das avaliações, na análise histórica e técnica de seus pontos fortes e fracos, e na análise de custos.

Já a norma ISO/IEC 15504 tem por objetivo principal prover uma infra-estrutura para avaliação de processos, podendo ser aplicada para uma organização entender o estado de seus processos com vistas a trabalhar a melhoria dos mesmos. Ela é composta por 5 partes, a saber:

- Parte 1: Conceitos e Vocabulário (Informativa) – estabelece uma introdução geral sobre os conceitos de avaliação de processos e um glossário para os termos relacionados;
- Parte 2: Execução de uma avaliação (Normativo) – define os requisitos mínimos para execução de uma avaliação que garanta consistência e repetição dos processos;
- Parte 3: Guia para execução de uma avaliação (Informativo) – estabelece uma interpretação dos requisitos para execução de uma avaliação;
- Parte 4: Guia para utilização em processos de melhoria e na determinação da capacidade de processos (Informativo) – identifica a avaliação de processos como uma atividade a ser executada como parte de uma iniciativa de melhoria de processos ou como parte de uma abordagem de determinação de capacidade.
- Parte 5: Um exemplo de um modelo de avaliação de processos (Informativo) – contém um exemplo de um Modelo de Avaliação de Processo baseado no

Modelo de Referência de Processo definido na ISO/IEC 12207, emendas 1 e 2.

Especificamente em sua parte 4, a ISO 15504 provê orientações sobre como usar avaliações dentro de um programa de melhoria de processos. A avaliação de processos segundo a ISO/IEC 15504 é baseada em um modelo bi-dimensional:

- a Dimensão de Processo é constituída por processos de ciclo de vida definidos no Modelo de Referência de Processos, o qual define um conjunto de processos caracterizados por uma descrição dos objetivos e dos resultados. Na Parte 5 dessa norma, um exemplo de modelo de referência é derivado diretamente da norma ISO/IEC 12207 emendas 1 e 2.
- a Dimensão de Capacidade consiste numa plataforma de medição composta por seis níveis de capacidade e pelos atributos de processo associados a cada nível. O resultado da avaliação consiste na determinação do perfil de cada um dos processos através da classificação dos respectivos atributos.

#### **2.4.2 - Avaliação e Melhoria de Processos no CMMI**

O CMMI (*Capability Maturity Model Integration*) (SEI, 2006a) surgiu com o intuito de integrar os vários modelos CMM independentes, resolvendo assim, o problema de se utilizar modelos que possuíam diferenças em sua arquitetura, conteúdo e abordagem. Desse modo, ele foi desenvolvido inicialmente baseado na evolução e integração de três modelos: a versão 2.0 do SW-CMM (*Capability Maturity Model for Software*), o SE-CMM: EIA 731 (*System Engineering Capability Maturity Model*) e o IPD-CMM (*Integrated Product Development Capability Maturity Model*) (SALVIANO, 2006).

O CMMI é um modelo estruturado em termos de áreas de processo (*Process Area – PA*), que consistem em práticas relacionadas que coletivamente satisfazem um conjunto de objetivos. Ele também oferece duas abordagens diferentes para a melhoria de processos, sendo conhecidas por “modelo em estágios” e “modelo contínuo”.

A representação por estágios prescreve a ordem para implementação de cada área de processo de acordo com níveis de maturidade, definindo um caminho de melhoria para uma organização do nível Inicial (1) para o nível Otimizado (5). Atingindo um nível de maturidade, garante-se uma base adequada para buscar o próximo estágio.

A representação contínua oferece uma abordagem flexível para melhoria de processos. Uma organização pode escolher melhorar a qualidade de um processo específico ou trabalhar em diversas áreas de forma alinhada aos objetivos de negócio. Os níveis de capacidade são usados para medir uma área, indo de um processo não executado para um processo otimizado.

Especificamente em relação à melhoria de processos, o CMMI define em seu nível 3 de maturidade a área de processo Foco no Processo Organizacional, cujo objetivo é planejar, implementar e implantar melhorias nos processos organizacionais, tomando por base um entendimento abrangente dos pontos fortes e fracos dos atuais processos e ativos de processos da organização.

Em relação à avaliação de processos tendo por base o CMMI, pode-se utilizar o Método de Avaliação CMMI Padrão para Melhoria de Processo (*Standard CMMI Appraisal Method for Process Improvement - SCAMPI*) (SEI, 2006b), que consiste em agrupar evidências objetivas que são coletadas pelo uso de instrumentos, apresentações, documentos, reuniões e entrevistas, possibilitando, assim, a detecção de pontos fortes e pontos fracos de uma organização, além de poder ser usado para fornecer uma classificação referente ao nível CMMI em que a organização se encontra. Essa classificação se dá após a transformação dos dados em informações objetivas e a análise delas em relação ao atendimento e aderência das práticas solicitadas por esse modelo.

### **2.4.3 - Avaliação e Melhoria de Processos no MPS.BR**

O MPS.BR (Melhoria de Processo do Software Brasileiro) (SOFTEX, 2007a) é um modelo para avaliação e melhoria de processos de software, tendo como base técnica as normas ISO/IEC 12207 e suas emendas 1 e 2, e a ISO/IEC 15504, além de cobrir todo o conteúdo do modelo CMMI. Ele possui três componentes, a saber: (i) *Modelo de Referência (MR-MPS)*, que contém os requisitos a serem atendidos pelas organizações, bem como os processos e os níveis de maturidade; (ii) *Método de Avaliação (MA-MPS)*, que contém o processo de avaliação, os requisitos para os avaliadores e os requisitos para averiguação da conformidade ao MR-MPS; e (iii) *Modelo de Negócio (MN-MPS)*, que contém as regras para implantação do MPS.BR pelas empresas de consultoria, de software e avaliação.

O MPS.BR está descrito por meio de documentos em formato de guias (SOFTEX, 2007a):

- **Guia Geral:** contém a descrição geral do MPS.BR e detalha o Modelo de Referência (MR-MPS), seus componentes e as definições comuns necessárias para seu entendimento e aplicação;
- **Guia de Aquisição:** descreve um processo de aquisição de software e serviços correlatos. É descrito como forma de apoiar as instituições que queiram adquirir produtos de software e serviços correlatos apoiando-se no MR-MPS;
- **Guia de Avaliação:** descreve o processo e o método de avaliação MA-MPS, os requisitos para avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IAs);
- **Guia de Implementação:** composto de 7 partes, cada uma delas descrevendo como implementar um determinado nível do MR-MPS.

O Modelo de Referência MR-MPS define sete níveis de maturidade, combinando processos e capacidade de processos. Os níveis de maturidade estabelecem patamares de evolução de processo, caracterizando estágios de melhoria de implementação de processos na organização. Os níveis de maturidade do MPS.BR são os seguintes: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A.

A capacidade do processo é um conjunto de atributos de processo descrito em termos de resultados. A capacidade estabelece o grau de refinamento e institucionalização com que o processo é executado na organização. À medida que evolui nos níveis, um maior ganho de capacidade para desempenhar o processo é atingido pela organização. A capacidade do processo possui nove atributos de processo:

- AP 1.1 – O processo é executado;
- AP 2.1 – O processo é gerenciado;
- AP 2.2 – Os produtos de trabalho do processo são gerenciados;
- AP 3.1 – O processo é definido;
- AP 3.2 – O processo está implementado;
- AP 4.1 – O processo é medido;
- AP 4.2 – O processo é controlado;
- AP 5.1 – O processo é objeto de inovações;
- AP 5.2 – O processo é otimizado continuamente.

Em relação à melhoria de processos, o MPS.BR define, em seu nível E, o processo Avaliação e Melhoria do Processo Organizacional, cujo propósito é determinar o quanto os processos padrão da organização contribuem para alcançar os objetivos de negócio da organização e para apoiar a organização a planejar, realizar e implantar melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos.

Já em relação à avaliação, o Guia de Avaliação do MPS.BR (SOFTEX, 2007b) descreve o processo e o Método de Avaliação MA-MPS, sendo baseado na norma ISO/IEC 15504-2:2003 (ISO/IEC, 2003). O propósito do Método de Avaliação MA-MPS é verificar a maturidade da unidade organizacional na execução de seus processos de software. Por sua vez, o processo de avaliação descreve o conjunto de atividades e tarefas a serem realizadas para atingir este propósito.

O processo e o Método de Avaliação MA-MPS foram definidos de forma a:

- Permitir a avaliação objetiva dos processos de software de uma organização/unidade organizacional;
- Permitir a atribuição de um nível de maturidade do MR-MPS com base no resultado da avaliação;
- Ser aplicável a qualquer domínio de aplicação na indústria de software;
- Ser aplicável a organizações /unidades organizacionais de qualquer tamanho.

Numa avaliação MPS são avaliados os resultados esperados de processo (REP) e também os resultados de atributos de processo (RAP) para cada um dos processos que fazem parte de um determinado nível, como por exemplo, os processos Gerência de Projetos e Gerência de Requisitos que compõem o nível G do MPS.BR.

Essa avaliação consiste em atribuir para cada um dos REPs e RAPs do processo sendo avaliado o grau de implementação, que pode ser: T – Totalmente Implementado, L – Largamente Implementado, P – Parcialmente Implementado, N – Não Implementado, NA – Não Avaliado ou F – Fora de escopo. Essa atribuição deve ocorrer inicialmente em relação a cada um dos projetos selecionados para serem avaliados. Em seguida, com base no grau de cada projeto, é calculado o grau de implementação do processo em nível organizacional.

Uma vez realizada a avaliação de todos os REPs e RAPs dos processos de um determinado nível, cabe à equipe de avaliação computar o grau de implementação dos processos da organização avaliada. A partir da definição desse grau de implementação,

o resultado final da avaliação, indicando se a organização alcançou ou não o nível MPS desejado, é apresentado pela equipe de avaliação para o representante da organização avaliada.

## ***2.5 Apoio Ferramental para a Avaliação e Melhoria de Processos de Software***

Pelo exposto nas seções anteriores, pode-se perceber que o processos de Avaliação e Melhoria de Processos (AMP) está fortemente relacionado a outros processos do ciclo de vida de software, com destaque para o processo de medição. Além disso, ele é um processo complexo e, portanto, prover apoio automatizado a ele é fundamental para que uma organização possa trabalhar a melhoria contínua de seus processos.

Existem diversos trabalhos tratando do apoio automatizado ao processo de AMP relatados na literatura, com destaque para a avaliação de processos. A seguir, alguns deles são brevemente discutidos.

### **2.5.1 – Ferramenta APPRAISAL ASSISTANT**

A ferramenta *Appraisal Assistant* (SQI, 2007) foi desenvolvida pelo *Software Quality Institute – SQI* da *Griffth University* com o objetivo de apoiar avaliações de capacitação de processo ou de maturidade organizacional. Ela segue os requisitos dos processos de avaliação tanto da ISO/IEC 15504 como também do SCAMPI.

Dentre suas principais características podem-se citar: (i) suporte ao método de avaliação SCAMPI e ao processo de avaliação definido pela norma ISO/IEC 15504, em sua parte 2; (ii) conversão de resultados de diferentes modelos, como, por exemplo, os dois acima citados; (iii) suporte a medidas para acompanhamento dos processos de avaliações; (iv) apoio à rastreabilidade entre os artefatos avaliados; e (v) apoio à geração de relatórios da avaliação de processo.

### **2.5.2 – Ferramenta ProEvaluator**

A ProEvaluator (XAVIER, 2007) é uma ferramenta que provê apoio ao processo de avaliação de processos. Mais especificamente, ela foca no apoio à avaliação dentro do contexto do MPS.BR, apoiando as atividades do processo de avaliação referentes ao Método de Avaliação do MPS.BR (MA-MPS).

Dentre suas principais características podem-se citar: (i) ser adaptável a organizações de diferentes domínios e tamanhos; (ii) ser totalmente baseada no modelo de qualidade MPS.BR; (iii) oferecer relatórios parciais durante a realização da avaliação e um relatório final com o resultado da avaliação; e (iv) ser uma ferramenta livre e disponível para download na internet.

### **2.5.3 – Ferramenta PISA**

A ferramenta PISA (FABRINI, 2003, apud XAVIER, 2007) provê apoio para as atividades de avaliação de processo de software baseadas na norma ISO/IEC 15504. Ela oferece apoio às atividades de coleta e validação de dados e atribuição de graus para os atributos de processo.

A ferramenta apóia quatro fases do processo de avaliação, a saber: (i) *preparação*, fornecendo apoio às atividades de registro de dados da organização, configuração dos pesos das características da avaliação e registro do escopo da avaliação; (ii) *coleta de dados*, apoiando o preenchimento de *checklists*; (iii) *avaliação*, apoiando a realização das atividades de análise de práticas base e de gerenciamento; e, por fim, (iv) *emissão de relatórios*, gerando o relatório final da avaliação.

### **2.5.4 – Ferramenta FAMP**

A ferramenta FAMP (SIQUARA, 2007) provê apoio automatizado para a realização das atividades de avaliação de processo de software baseadas no modelo CMMI, atendendo, assim, às orientações provenientes do método de avaliação SCAMPI. Ela proporciona suporte à identificação dos pontos fortes e fracos de um processo e, além disso, apóia o acompanhamento dos planos de ação elaborados durante a avaliação de um processo.

Suas principais características são: (i) definição do escopo de uma avaliação; (ii) identificação dos papéis das pessoas envolvidas na avaliação; (iii) apoio à rastreabilidade entre as evidências objetivas da avaliação; e (iv) geração de base histórica que pode ser consultada em avaliações futuras.

### **2.5.5 – Ferramenta AvalPro**

A ferramenta AvalPro (ANDRADE, 2005) apóia as atividades do grupo de Garantia da Qualidade do Processo e do Produto (GQPP) da organização. Seu principal objetivo é apoiar o grupo de GQPP na avaliação de processos instanciados em projetos



da organização. A ferramenta AvalPro é parte integrante da Estação TABA, um ambiente de desenvolvimento de software centrado em processo e orientado a organização (VILELLA et al., 2004).

Dentre suas principais funcionalidades, destacam-se: (i) planejar a avaliação do processo; (ii) avaliar processos; (iii) realizar avaliação post-mortem; (iv) calcular medidas para monitoração dos processos; (v) analisar resultados preliminares; (vi) realizar reunião de consenso; (vii) elaborar o relatório de avaliação do processo instanciado; e (viii) comunicar resultados da avaliação.

### **2.5.6 – Ferramenta PILOT**

A ferramenta Pilot (SILVA FILHO et al., 2007) apóia a realização de avaliação de propostas de melhoria de um processo de forma sistemática, planejada e controlada em projetos-piloto para que essas melhorias possam ser observadas e mensuradas antes de serem institucionalizadas dentro de uma organização. A ferramenta Pilot é também parte integrante da Estação TABA (VILELLA et al., 2004).

Dentre as principais funcionalidades de Pilot podem ser citadas: (i) a definição do projeto-piloto; (ii) a definição das diretrizes para a realização do projeto-piloto; (iii) a visualização do plano de execução do projeto-piloto; (iv) o monitoramento do andamento do projeto-piloto; (v) a análise dos resultados dos projetos-piloto; e (vi) a visualização dos resultados da análise.

### **2.5.7 – Integração de Ferramentas de Apoio ao Processo de Avaliação e Melhoria de Processos**

Como pode ser observado pelas ferramentas citadas anteriormente, há um grande número de ferramentas que trabalham de forma isolada para apoiar uma parte do processo de AMP e, com isso, acabam por apoiar apenas parcialmente o processo de avaliação e melhoria de processos. Uma forma de prover essa integração é pensar em desenvolver ferramentas dentro de Ambientes de Desenvolvimento de Software (ADSs), como é o caso da ferramenta Pilot.

De maneira geral, os ADSs, pelo seu caráter de integração, se apresentam como uma opção que permite uma ligação forte entre ferramentas de apoio à Engenharia de Software. O principal objetivo de um ADS é prover funcionalidades integradas para apoiar a realização de grande parte das atividades de um projeto de software, sendo sua principal característica a integração de ferramentas. Por meio dessa integração, os ADSs

são capazes de manter um alto grau de controle do que está sendo desenvolvido dentro de seu contexto.

Em relação ao processo AMP, e seus relacionamentos com outros processos, é muito importante que haja essa integração, pois as informações advindas principalmente dos processos de medição e avaliação são base para que melhorias em um processo possam ser realizadas.

## ***2.6 Considerações Finais do Capítulo***

Neste capítulo foram apresentados os conceitos relativos ao domínio de Avaliação e Melhoria de Processos de Software, tema foco desta dissertação, com o intuito de fornecer ao leitor um panorama geral sobre o assunto e uma base teórica suficiente para a leitura dos próximos capítulos deste trabalho.

Além disso, também foram discutidas questões específicas de como os principais modelos e normas de qualidade tratam o processo de AMP, tendo sido um deles, mais especificamente o MPS.BR, a base para a definição de um processo de AMP apresentado no Capítulo 3 desta dissertação.

Também foi promovida uma discussão sobre o apoio automatizado para o processo de AMP. Foi evidenciada a dificuldade em prover apoio automatizado e integrado entre as várias ferramentas que apóiam o processo de AMP e seus relacionamentos. Desse modo, com o intuito de prover uma base consistente para apoiar o desenvolvimento de ferramentas de apoio ao domínio de AMP foi desenvolvida uma ontologia para esse domínio apresentada no Capítulo 4. Por fim, para prover apoio automatizado, mesmo que parcial, ao processo de AMP definido no Capítulo 3, são apresentadas no Capítulo 5 as ferramentas desenvolvidas no contexto do ambiente ODE (*Ontology-based software Development Environment*) (FALBO et al., 2003), um ambiente de desenvolvimento de software baseado em ontologias.

# Um Processo de Avaliação e Melhoria de Processos de Software

O objetivo deste capítulo é definir um processo de Avaliação e Melhoria de Processos de Software (AMP) que seja baseado no modelo de qualidade MPS.BR, apresentando também a interação entre o processo de AMP proposto e outros processos do ciclo de vida de software. Além disso, também é objetivo deste capítulo discutir funcionalidades para apoiar o processo de AMP definido. Desse modo, a abordagem adotada para este capítulo se dá por: (i) a apresentação do processo de AMP definido; (ii) uma discussão sobre as interações entre o processo de AMP e outros processos; e por fim (iii) uma discussão sobre funcionalidades necessárias para prover apoio automatizado ao processo de AMP proposto.

### 3.1 Introdução

O domínio referente à Avaliação e Melhoria de Processos de Software (AMP) é bastante amplo e complexo. Dessa maneira, uma organização que busca a melhoria contínua de seus processos necessita despender um esforço considerável para alcançar seu objetivo, que é a melhoria contínua de seus processos e, por conseguinte, alcançar melhores resultados no desenvolvimento de produtos de software.

Para tal, é importante que a organização defina um processo de melhoria de processos. Contudo, essa definição não é trivial de ser realizada, uma vez que há muitas variáveis críticas envolvidas na melhoria de processos, dentre elas: (i) a cultura organizacional, (ii) o apoio da alta gerência, (iii) a dificuldade em automatizar o processo e (iv) a existência de vários modelos de processo, o que dificulta a escolha do que melhor se ajusta à realidade da organização.

Dado que o escopo deste trabalho é prover apoio automatizado para a avaliação e melhoria de processos de software e tendo em mente os pontos críticos citados acima, sobretudo (iii) e (iv), decidiu-se definir um processo de AMP geral que seja baseado em um modelo de qualidade reconhecido nacionalmente no mercado de software, o MPS.BR (SOFTEX, 2007a), uma vez que, com isso, há uma indicação de que boas práticas estão presentes nesse processo definido. É importante ressaltar que essa escolha levou em conta, ainda, o fato do MPS.BR ser baseado em normas e modelos

internacionais de qualidade, a saber: CMMI (SEI, 2006a), ISO/IEC 12207 (ISO/IEC, 2008) e ISO/IEC 15504 (ISO/IEC, 2003).

Uma vez definido o processo de avaliação e melhoria de processos de software, fica mais fácil prover apoio automatizado para ele, pois já se sabe a priori quais são as atividades e relacionamentos que devem ser apoiados pelas funcionalidades das ferramentas a serem desenvolvidas.

Vale ressaltar que o apoio automatizado ao processo de AMP é fator fundamental para sua execução e sucesso, uma vez que, conforme citado anteriormente, a melhoria contínua de processos não é tarefa trivial de ser implementada, requerendo grandes esforços e custos financeiros para as organizações.

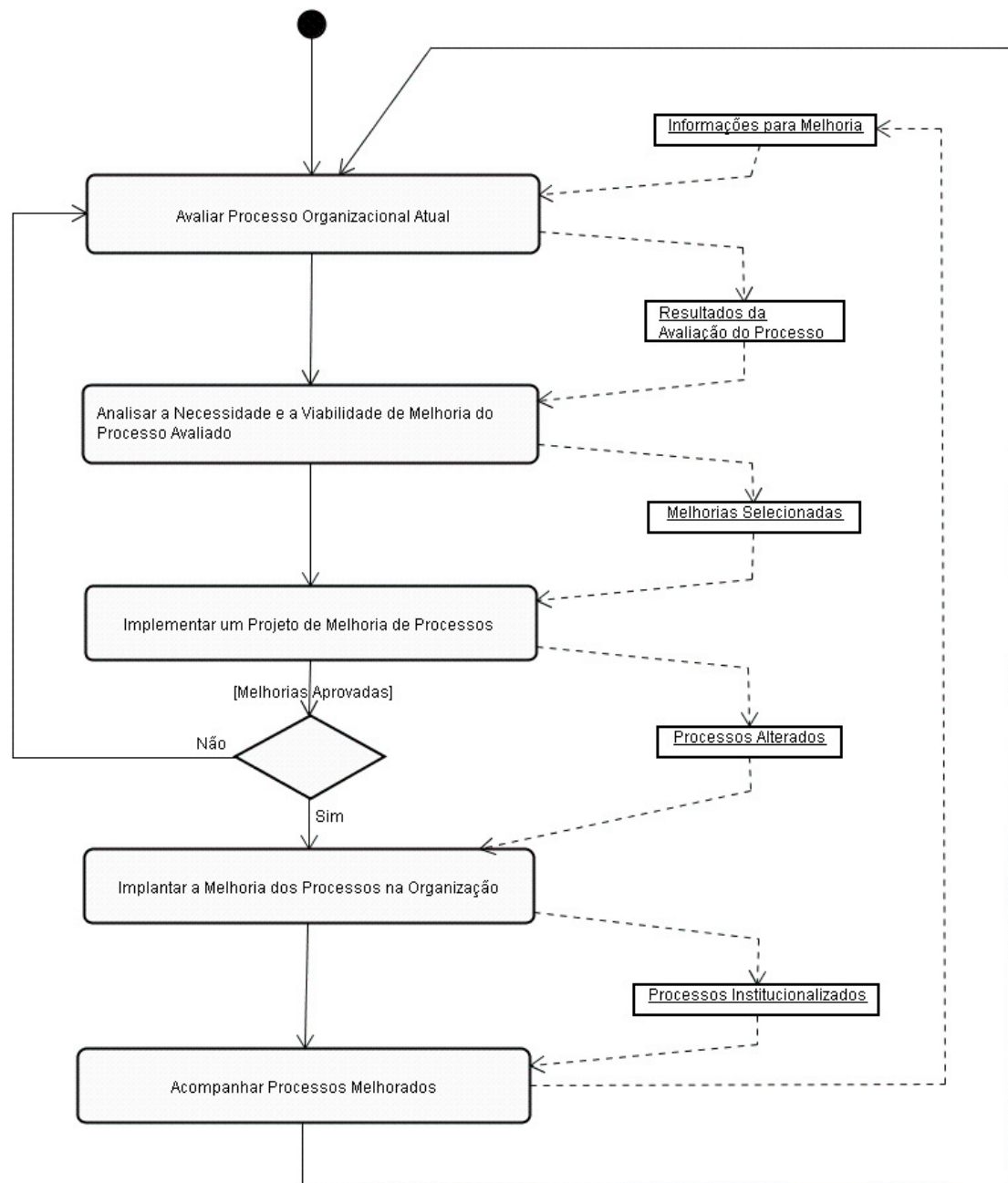
Este capítulo está estruturado da seguinte forma: a Seção 3.2 apresenta a definição do processo de AMP proposto; a Seção 3.3 fornece uma visão geral sobre a interação entre o processo de AMP definido e outros processos do ciclo de vida de software; na Seção 3.4 discutem-se funcionalidades requeridas para apoiar de forma automatizada as atividades do processo de AMP definido; por fim, a Seção 3.5 apresenta as considerações finais do capítulo.

### ***3.2 Definição do Processo de AMP***

Há na literatura muitas informações acerca do tema avaliação e melhoria de processo de software uma vez que esse domínio possui uma amplitude e complexidade considerável. Contudo, os principais trabalhos que serviram de fonte para a definição do processo de AMP proposto foram (PFLLEGER, 2004), (POTTER e SAKRY, 2002), (ZAHARAN, 1998), (McGARRY et al., 2002) e (FLORAC e CARLETON, 2000). Além disso, também foram consultadas as normas internacionais ISO/IEC 12207 (ISO/IEC, 2008) e ISO/IEC 15504 (ISO/IEC, 2003) e, também, os modelos de qualidade CMMI (SEI, 2006a) e MPS.BR (SOFTEX, 2007a).

A Figura 3.1 mostra um diagrama de atividades apresentando a interação entre as cinco principais atividades do processo de AMP definido. Pela análise desse diagrama, percebe-se que o processo de AMP definido é cíclico em duas instâncias. Inicialmente, apenas melhorias aprovadas são institucionalizadas na organização. As melhorias que não são aprovadas durante o projeto piloto devem voltar à fase inicial, iniciando novamente o processo. Uma vez institucionalizadas as melhorias, os processos continuam sendo acompanhados com o intuito de garantir que elas estão sendo

cumpridas na organização e, também, para que novas sugestões de melhoria possam ser observadas.



**Figura 3.1 – Diagrama de Atividades do Processo de AMP Definido.**

São os seguintes os objetivos das macro-atividades propostas:

A1 *Avaliar Processo Organizacional Atual*: tem por objetivo avaliar um processo organizacional em seu estado atual e registrar os resultados alcançados com essa avaliação. Esses resultados servirão de base para a organização definir a necessidade e a viabilidade de uma possível melhoria do processo avaliado;

- A2 *Analisar a Necessidade e a Viabilidade da Melhoria do Processo Avaliado*: visa analisar os resultados da avaliação do processo, identificar o propósito da melhoria do processo avaliado, identificar os riscos e impactos gerados pela melhoria do processo e, por fim, com base nessas informações, tomar a decisão sobre a necessidade e a viabilidade de inserir a melhoria do processo no conjunto de melhorias de processo que serão realizadas no contexto de um projeto de melhorias de processo organizacional;
- A3 *Implementar um Projeto de Ações de Melhoria de Processos*: tem por objetivo realizar a implementação de um projeto de ações de melhoria de processos, incluindo planejamento, execução, monitoramento e controle, avaliação dos artefatos gerados e, também, a implantação das melhorias em projetos piloto, com o intuito de avaliar sua eficiência e eficácia em relação à proposta que originou a necessidade de melhoria;
- A4 *Institucionalizar a Melhoria dos Processos na Organização*: sua meta é, uma vez avaliada a melhoria de processo durante o projeto de melhoria de processos, institucionalizar essa melhoria na organização. Essa institucionalização implica em implantar a melhoria em âmbito organizacional, inclusive com o treinamento dos envolvidos e, também, registrar e publicar as lições aprendidas com a melhoria de processo em questão;
- A5 *Acompanhar os Processos Melhorados*: visa acompanhar o uso dos processos institucionalizados na organização, de modo a detectar novas propostas de melhoria, o que mostra que o processo de AMP definido é cíclico e, sendo assim, preocupa-se com a melhoria contínua dos processos da organização.

Durante a definição do processo de AMP proposto não houve uma preocupação com a definição de artefatos, procedimentos e recursos para as atividades do processo. Cabe à organização definir esses ativos, ainda que a Figura 3.1 aponte as principais informações que os artefatos produzidos por essas atividades devem prover. O foco principal foi definir quais são as atividades que devem ser realizadas para que uma melhoria de processo ocorra dentro de uma organização. A Tabela 3.1 apresenta o detalhamento do processo proposto em termos de sub-atividades.

**Tabela 3.1 – Atividades do Processo de AMP Definido**

Identificação	Atividade
<b>A1</b>	<b>Avaliar Processo Organizacional Atual</b>
A1.1	Coletar e avaliar informações úteis para a melhoria do processo
A1.2	Elaborar uma proposta de escopo para a avaliação do processo
A1.3	Definir critérios para a avaliação do processo
A1.4	Avaliar o processo segundo os critérios definidos
A1.5	Identificar pontos fortes, pontos fracos e oportunidades de melhoria do processo
A1.6	Registrar os resultados da avaliação do processo
<b>A2</b>	<b>Analisar a Necessidade e Viabilidade da Melhoria do Processo Avaliado</b>
A2.1	Analisar o resultado da avaliação do processo
A2.2	Definir o propósito da melhoria do processo
A2.2.1	Identificar os objetivos da melhoria do processo
A2.2.2	Identificar os problemas a serem resolvidos com a melhoria do processo
A2.2.3	Identificar os benefícios a serem gerados com a melhoria do processo
A2.3	Analisar os riscos e o impacto da melhoria do processo
A2.3.1	Identificar e analisar os riscos organizacionais da melhoria do processo
A2.3.2	Analisar o impacto da melhoria do processo para a organização
A2.3.3	Analisar o impacto da melhoria do processo para as pessoas envolvidas
A2.4	Identificar os benefícios a serem gerados com a melhoria do processo
<b>A3</b>	<b>Implementar um Projeto de Ações de Melhoria de Processos</b>
A3.1	Planejar o projeto de ações de melhoria de processos
A3.2	Executar o projeto de ações de melhoria de processos
A3.3	Verificar, validar e monitorar o andamento do projeto de ações de melhoria de processos
A3.4	Avaliar os produtos gerados pelo projeto de ações de melhoria de processos
A3.5	Implantar a melhoria de processos primeiramente em um projeto piloto, incluindo o treinamento dos envolvidos
A3.6	Avaliar o resultado da melhoria de processos no projeto piloto
<b>A4</b>	<b>Institucionalizar a Melhoria dos Processos na Organização</b>
A4.1	Implantar a melhoria de processos na organização, incluindo o treinamento dos envolvidos
A4.2	Publicar os resultados e lições aprendidas alcançados com a melhoria de processos
<b>A5</b>	<b>Acompanhar os Processos Melhorados</b>

### 3.3 Interação entre o Processo de AMP e outros Processos

A amplitude e a complexidade do processo de AMP podem ser observadas tanto pelas atividades que o compõem quanto pelas relações com outros processos do ciclo de vida do software.

A Figura 3.2 apresenta, de forma geral, como se dá a interação entre o processo de AMP e outros processos do MPS.BR, mostrando as informações principais trocadas entre eles e procurando destacar, ainda, em que nível de maturidade do MPS.BR cada um dos processos se encontra. As siglas nessa figura designam os respectivos processos no MPS.BR, conforme Tabela 3.2, que apresenta, ainda, o propósito de cada um deles.

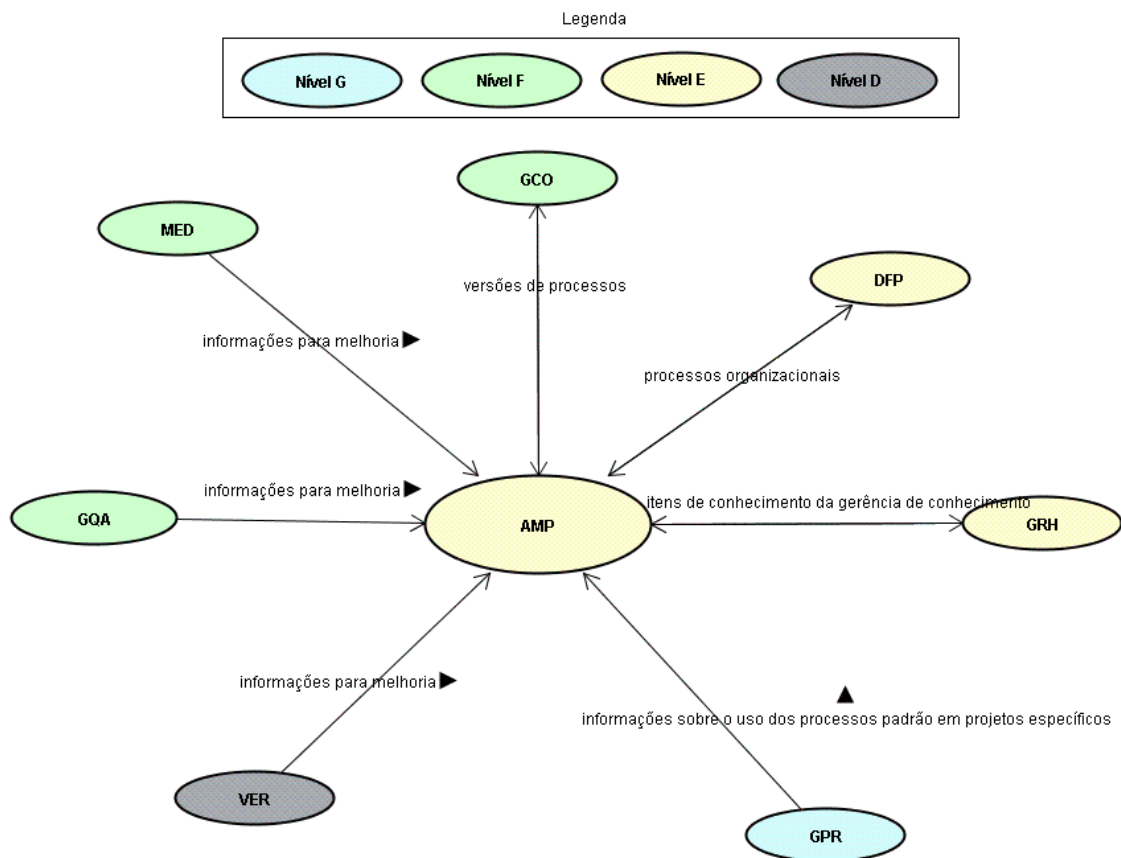


Figura 3.2 – Interação entre o processo de AMP definido e outros processos MPS.BR.



**Tabela 3.2 – Identificação dos Processos do MPS.BR.**

<b>Sigla do MPS.BR</b>	<b>Nome do Processo</b>	<b>Propósito do Processo</b>
GPR	Gerência de Projeto	O propósito do processo Gerência de Projetos é estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto.
MED	Medição	O propósito do processo Medição é coletar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais.
GQA	Garantia da Qualidade	O propósito do processo Garantia da Qualidade é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos e recursos predefinidos.
GCO	Gerência de Configuração	O propósito do processo Gerência de Configuração é estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos.
AMP	Avaliação e Melhoria do Processo Organizacional	O propósito do processo Avaliação e Melhoria do Processo Organizacional é determinar o quanto os processos padrão da organização contribuem para alcançar os objetivos de negócio da organização e para apoiar a organização a planejar, realizar e implantar melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos.
DFP	Definição do Processo Organizacional	O propósito do processo Definição do Processo Organizacional é estabelecer e manter um conjunto de ativos de processo organizacional e padrões do ambiente de trabalho usáveis e aplicáveis às necessidades de negócio da organização.
GRH	Gerência de Recursos Humanos	O propósito do processo Gerência de Recursos Humanos é prover a organização e os projetos com os recursos humanos necessários e manter suas competências consistentes com as necessidades do negócio.
VER	Verificação	O propósito do processo Verificação é confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados.

Há basicamente dois tipos principais de interação entre os processos do MPS.BR mostrados na figura e o processo de AMP. O primeiro diz respeito a processos que são executados em atividades de projetos organizacionais, provendo informações que são importantes insumos para a melhoria de processos da organização, são eles: GQA, VER, MED e GPR. Já o segundo tipo de interação diz respeito a processos que definem e apóiam a infra-estrutura de processos na organização. Esses, por sua vez, servem de base para a melhoria e devem ser consultados e modificados sempre que uma determinada melhoria de processo aconteça, o que é o caso dos processos DFP, GCO e GRH.

O processo de GQA provê importantes informações para a melhoria de processos, haja vista que durante a realização de atividades de garantia da qualidade em projetos específicos, em que se avalia a aderência a padrões, podem-se identificar não conformidades que ocorrem sistematicamente em vários projetos e, portanto, pode-se perceber que os padrões organizacionais não estão satisfatórios para atender às necessidades dos projetos.

Analogamente, durante a realização das atividades de verificação de projetos da organização, tal como revisões por pares, também podem ser identificados problemas que ocorrem sistematicamente nos projetos, apontando a necessidade de uma possível melhoria no mesmo.

O processo de medição provê informações básicas para as atividades de avaliação de processos organizacionais. Durante a realização de suas atividades, são obtidas informações objetivas acerca do desempenho dos seus processos e da qualidade dos produtos produzidos pelo mesmo, usadas nas avaliações de processos.

Por fim, a interação entre os processos de GPR e de AMP é bastante natural, pois durante a gerência de projetos podem-se perceber atividades que são gargalos no processo, consumindo muitos recursos ou sistematicamente apresentando problemas de prazos e custos, informações essas valiosas para a avaliação de processos e, por conseguinte, para apontar melhorias em processos da organização.

Em relação ao segundo tipo de interação entre processos, o processo de AMP avalia e melhora processos organizacionais que foram definidos pelo processo de DFP. Após uma melhoria, é necessário que os ativos de processo alterados sejam institucionalizados e mantidos na biblioteca de ativos da organização, o que é de responsabilidade do processo de DFP.

Uma vez que o processo de AMP altera processos e ativos de processo, é necessário controlar a versão desses elementos. Assim, o processo de AMP interage diretamente com o processo de GCO, responsável pelo controle de alterações e de versões.

Por fim, o processo de GRH, mais especificamente em relação à gerência de conhecimento da organização, serve de base para que informações sobre itens de conhecimento da organização apoiem a decisão sobre uma melhoria de processo. Além disso, após uma melhoria ter sido realizada, lições aprendidas podem ser inseridas no sistema de gerência de conhecimento para que sirvam de fonte de pesquisa em futuras melhorias de processo da organização.

### ***3.4 Funcionalidades para Apoiar o Processo de AMP Definido***

Uma vez definido o processo de AMP e discutidas suas interações com os demais processos organizacionais e do ciclo de vida de um software, torna-se necessária a identificação e discussão de funcionalidades que podem prover apoio automatizado ao processo de AMP definido. A seguir são apresentadas funcionalidades que foram classificadas em dois grupos, a saber: gerais e específicas. As funcionalidades gerais tratam do apoio a processos para vários tipos de projetos, como, por exemplo, o apoio à definição de um processo. Mais ainda, por serem gerais, também se aplicam ao processo de AMP definido. Já as específicas dizem respeito, a priori, especificamente ao apoio automatizado ao processo de AMP definido.

#### **3.4.1 Funcionalidades Gerais para Apoiar o Processo de AMP Definido**

As funcionalidades gerais identificadas para prover apoio automatizado ao processo de AMP definido são:

- **FG1 – Apoio à definição do processo de avaliação e melhoria de processos:**  
Essa funcionalidade deve permitir que sejam definidos os processos que apoiem a avaliação e melhoria de processos, incluindo o processo de AMP. Ela deve contemplar (i) a definição das atividades, abrangendo precedências entre elas e uma decomposição em sub-atividades, (ii) a definição de artefatos gerados e consumidos por uma atividade, (iii) a alocação de recursos humanos e materiais (software e hardware) para a realização de uma atividade, e (iv) a definição dos procedimentos adotados na realização de uma atividade. A seguir são descritos

alguns exemplos de processos que devem ser contemplados por essa funcionalidade:

- FG1.1 – Apoio à definição do processo de AMP;
  - FG1.2 – Apoio à definição do processo de realização de um projeto de ações de melhoria de processos;
  - FG1.3 – Apoio à definição do processo de realização do projeto piloto para avaliação das melhorias realizadas.
- 
- **FG2 – Apoio à gerência de projetos relacionados com a melhoria (análogo a FG1):** Essa funcionalidade deve apoiar a elaboração dos planos do projeto e também o registro do controle da execução e realização de projetos relacionados com a melhoria de processos.
  - **FG3 – Apoio à realização de projetos relacionados com a melhoria (análogo a FG1):** Essa funcionalidade deve permitir que sejam registradas ações referentes aos processos de VER, VAL e GQA relacionadas ao acompanhamento do andamento do projeto.
  - **FG4 – Apoio à realização de treinamentos nos processos alterados:** Essa funcionalidade deve permitir, dentre outros, que seja registrado um plano de treinamento para as pessoas envolvidas com as melhorias de processo institucionalizadas. Além disso, também deve permitir que seja registrada a participação das pessoas envolvidas em cada um dos treinamentos realizados. Como a realização de treinamento deve ser parte do processo de GRH de uma organização, essa funcionalidade pode ser vista como funcionalidade de apoio ao processo de GRH da organização.

### **3.4.2 Funcionalidades Específicas para Apoiar o Processo de AMP Definido**

As funcionalidades específicas identificadas para prover apoio automatizado ao processo de AMP definido são:

- **FE1 – Apoio à coleta de informações que servirão de base para a melhoria de um processo:** Essa funcionalidade deve permitir que sejam registradas proposições de melhoria para um processo. Essas sugestões vêm tipicamente de processos relacionados ao acompanhamento e controle de projetos, dentre eles: MED, VAL, VER, GQA e GPR.

- **FE2 – Apoio à definição de critérios para a avaliação de processos e seus ativos:** Essa funcionalidade deve permitir que sejam cadastrados critérios organizacionais de avaliação de processos e seus ativos. Em um segundo momento (ver F5 e F11), esses critérios podem ser selecionados, por exemplo, por meio do cadastro de *checklists*, para avaliar tanto um processo em si, como partes dele, ou seja, um artefato, uma atividade ou outro elemento qualquer do processo a ser avaliado. Esse é um cadastro geral, ou seja, essa funcionalidade apenas cadastra critérios para cada um dos tipos de avaliação possíveis de um processo. Já a seleção desses critérios para serem aplicados em uma avaliação de processo é parte do escopo das funcionalidades F5 e F11.
- **FE3 – Apoio à elaboração de uma proposta de escopo para a avaliação de um processo:** Essa funcionalidade deve permitir que seja cadastrada uma proposta de escopo para a avaliação de um processo. Essa proposta deverá conter o processo a ser avaliado e ativos do processo que serão avaliados como parte da avaliação do processo (atividades, artefatos, etc).
- **FE4 – Apoio à avaliação de um processo segundo critérios definidos:** Essa funcionalidade deve permitir que sejam selecionados, dentre os critérios gerais previamente cadastrados, quais serão os critérios utilizados para a avaliação de um processo específico. Além disso, essa funcionalidade também deve apoiar a avaliação em si, como por exemplo, o preenchimento de um ou mais *checklists* para avaliar um processo específico.
- **FE5 – Registro de conclusões de uma avaliação de processo:** Essa funcionalidade deve permitir, a partir de uma avaliação de processo já realizada, que sejam registradas conclusões acerca dessa avaliação. Esses registros devem ser pontos fortes, pontos fracos e oportunidades de melhoria referentes ao processo avaliado.
- **FE6 – Apoio à análise da avaliação de um processo:** Essa funcionalidade deve permitir que sejam realizadas análises referentes à medição de características do processo avaliado.
- **FE7 – Apoio à definição da proposta de melhoria do processo:** Deve permitir que seja definida claramente qual será a proposta da melhoria do processo. Essa definição deve contemplar quais são os objetivos da melhoria, os

problemas a serem resolvidos e os benefícios gerados após a melhoria ser implementada na organização.

- **FE8 – Apoio à análise dos riscos da melhoria do processo:** Deve permitir que sejam identificados quais os riscos organizacionais relacionados com a melhoria do processo. Além da identificação, também deve permitir que os riscos sejam classificados e analisados quanto à probabilidade e o impacto e que sejam descritas ações de contingência e mitigação para cada um dos riscos identificados.
- **FE9 – Apoio à análise dos impactos organizacionais com a melhoria do processo:** Deve permitir que sejam registradas análises dos possíveis impactos para a organização dessa melhoria do processo. Essas análises podem ser baseadas na análise da avaliação do processo, na proposta de melhoria do processo e na análise de riscos identificados para essa melhoria do processo.
- **FE10 – Registro da necessidade e viabilidade da melhoria do processo:** Deve permitir que seja registrado um parecer final sobre a necessidade e viabilidade da melhoria do processo, ou seja, registrar a tomada de decisão por acrescentar ou não essa melhoria no escopo de um projeto de melhoria de processos. Essa tomada de decisão pode ser baseada na análise da avaliação do processo, na proposta de melhoria do processo, na análise de riscos identificados para essa melhoria do processo e na análise de impactos organizacionais para a melhoria do processo.
- **FE11 – Apoio à avaliação de produtos:** A avaliação de produtos deve seguir a mesma regra de avaliação de processo. Assim, essa funcionalidade também deve permitir que sejam selecionados, dentre os critérios gerais previamente cadastrados, quais serão os critérios utilizados para a avaliação de um produto específico. Além disso, essa funcionalidade também deve apoiar a avaliação em si, como por exemplo, o preenchimento de um ou mais *cheklists* para avaliar um produto específico.
- **FE12 – Apoio à atualização da biblioteca de processos organizacionais:** Essa funcionalidade deve permitir que seja atualizada a biblioteca de processos organizacionais no momento em que uma melhoria de processo é institucionalizada na organização. Essa atualização pode ser realizada por meio da execução do processo de GCO da organização.

- **FE13 – Apoio ao registro de resultados e lições aprendidas com a melhoria de processos:** Essa funcionalidade deve permitir que sejam registrados, por meio de parecer, por exemplo, os resultados finais sobre a melhoria de processos institucionalizada na organização. Além disso, também deve permitir que sejam registradas as lições aprendidas durante o processo de melhoria de processos realizado.

### ***3.5 Considerações Finais do Capítulo***

Neste capítulo foram apresentados o processo de AMP definido e os principais relacionamentos e interações com outros processos do ciclo de vida de software, tendo por base o modelo MPS.BR. Além disso, também foram identificadas e discutidas funcionalidades que formam a base para prover apoio automatizado ao processo de AMP definido. A partir de algumas dessas funcionalidades identificadas, foram desenvolvidas ferramentas para apoiar o processo de AMP definido, o que é apresentado no Capítulo 5 desta dissertação.

Devido à amplitude e complexidade do processo de AMP definido e suas fortes relações com outros processos, conforme apresentado neste capítulo, potencialmente há a necessidade de várias ferramentas para apoiá-lo. Mais ainda, essas ferramentas devem estar integradas de tal modo que funcionem adequadamente em conjunto. Para tal é importante que haja interoperabilidade entre elas, garantindo com isso que elas atribuem significados compatíveis para entidades do mundo real que fazem parte de seu domínio de interesse compartilhado. Uma forma de atender a esse requisito de interoperabilidade é a criação de uma ontologia que contemple o domínio de Avaliação e Melhoria de Processos. A ontologia em questão foi desenvolvida e será apresentada no próximo capítulo desta dissertação.

### **Uma Ontologia para o Domínio de Qualidade de Software com foco em Avaliação e Melhoria de Processos de Software**

O objetivo deste capítulo é apresentar a Ontologia de Qualidade de Software desenvolvida no contexto deste trabalho, mostrando a instanciação da mesma para as perspectivas de processo e produto de software. Desse modo, a abordagem adotada para este capítulo se dá por: (i) uma breve discussão sobre ontologias; (ii) uma apresentação do método SABiO usado na construção da ontologia de qualidade; (iii) a apresentação da ontologia de qualidade de software em si, organizada em sub-ontologias de Modelo de Qualidade, Medição e Avaliação; e, por fim, (iv) a instanciação dos conceitos dessa ontologia para processo e produto de software.

#### ***4.1 Introdução***

Dada a amplitude e complexidade do processo de Avaliação e Melhoria de Processos (AMP) e as fortes relações com outros processos, com destaque para a medição, prover apoio automatizado a esse processo requer, geralmente, diversas ferramentas. Mesmo quando se pensando em um ambiente integrado de apoio ao processo de AMP, deve-se levar em conta que as ferramentas que o compõem dificilmente serão desenvolvidas pela mesma equipe e com os mesmos interessados e, portanto, é muito provável que algumas dessas ferramentas sejam construídas tomando por base diferentes conceituações. Assim, para que essas ferramentas funcionem adequadamente em conjunto, deve-se garantir que elas atribuem significados compatíveis para entidades do mundo real que fazem parte de seu domínio de interesse compartilhado.

Quando se deseja integrar diferentes sistemas desenvolvidos independentemente, é fundamental ter uma representação precisa da conceituação do domínio de interesse dos mesmos (GUIZZARDI, 2005). Ontologias podem ser utilizadas para esse fim, servindo como uma interlíngua para se mapear conceitos usados em um sistema em outros, abrindo espaço para a troca e o compartilhamento de informações (JASPER e USCHOLD, 1999). Mais ainda, se tais sistemas forem construídos tomando por base a mesma ontologia, elas terão um potencial ainda maior para interoperar, em um cenário que JASPER e USCHOLD (1999) denominam de ontologias como especificação.



Pensando em ferramentas de apoio ao processo de AMP, é importante explicitar, mesmo que parcialmente, a conceituação envolvida no domínio de qualidade de software, o que pode ser feito com o uso de ontologias. Uma conceituação corresponde ao conjunto de conceitos usados para interligar abstrações de entidades de um dado domínio. Uma ontologia é uma descrição formal e explícita de uma conceituação (GUIZZARDI, 2005). Ao explicitar e formalmente definir os conceitos, relações, propriedades e restrições em um domínio particular, tem-se um artefato de engenharia que expressa a conceituação desse domínio, dita uma ontologia de domínio. Uma ontologia de domínio pode ter diversos usos, tais como (JASPER e USCHOLD, 1999): (i) apoiar a comunicação entre pessoas trabalhando no domínio; (ii) integração de dados, modelos e sistemas desenvolvidos de forma independente para o domínio, (iii) como uma especificação reutilizável para a construção de sistemas no domínio, (iv) como apoio para manutenção de sistemas, servindo de documentação dos sistemas desenvolvidos para o domínio etc. Pensando nesses usos, sobretudo em (iii), desenvolveu-se uma ontologia para o domínio de qualidade de software, com foco em avaliação e melhoria de processos, incluindo medição.

Este capítulo apresenta a Ontologia de Qualidade de Software desenvolvida, visando formalizar parcialmente o conhecimento envolvido nesse domínio. Ela foi desenvolvida com o propósito de estabelecer uma conceituação comum sobre esse domínio, de modo a apoiar a integração de ferramentas de apoio ao processo de AMP. Mais especificamente, a ontologia proposta foi usada como uma especificação base para apoiar o desenvolvimento integrado de ferramentas de apoio ao processo de AMP no contexto do Projeto ODE (*Ontology-based software Development Environment*) (FALBO et al., 2003), que visa ao desenvolvimento de um ambiente de desenvolvimento de software baseado em ontologias.

Assim, este capítulo está estruturado da seguinte forma: a Seção 4.2 discute brevemente ontologias; a Seção 4.3 apresenta o método de construção de ontologias SABiO, utilizado neste trabalho; a Seção 4.4 apresenta em detalhes a ontologia desenvolvida; a Seção 4.5 discute trabalhos relacionados; e por fim, a Seção 4.6 apresenta as considerações finais do capítulo.

## 4.2 Ontologias

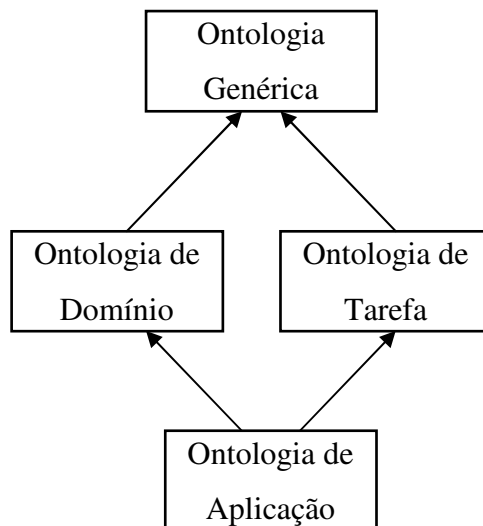
É crescente o uso de ontologias para estabelecer uma linguagem comum para o compartilhamento e a reutilização do conhecimento sobre domínios de aplicação durante o desenvolvimento de software (COTA et al., 2004). Uma ontologia define um vocabulário específico usado para descrever certa realidade, mais um conjunto de decisões explícitas fixando de forma rigorosa o significado pretendido para o vocabulário. Uma ontologia envolve, então, um vocabulário de representação que captura os conceitos e relações em algum domínio e um conjunto de axiomas, que restringem a sua interpretação (GUARINO, 1998).

Segundo JASPER e USCHOLD (1999), ontologia é o termo usado para se referir ao entendimento compartilhado em algum domínio de interesse, o qual pode ser usado como uma infra-estrutura unificada para resolver problemas, evitando a redescoberta de resultados equivalentes. Já para SPIN et al. (2002), ontologias, no contexto da Ciência da Computação, são recursos baseados em computador que representam semânticas compartilhadas sobre domínios.

Ainda que se tenha falado acima de ontologia de uma maneira mais específica, voltada para o conhecimento de domínio, pode-se destacar pelo menos quatro tipos de ontologias, segundo o que elas se propõem a modelar (GUARINO, 1998):

- **Ontologias Genéricas:** descrevem conceitos gerais, como espaço, tempo, problema, objeto, evento, ação etc.
- **Ontologias de Domínio:** descrevem o vocabulário relacionado a um domínio, como, por exemplo, medicina, direito etc.
- **Ontologias de Tarefa:** descrevem o vocabulário relacionado a uma tarefa genérica, como, por exemplo, diagnose, venda etc.
- **Ontologias de Aplicação:** descrevem conceitos dependentes de um domínio e de uma tarefa particulares, os quais são, freqüentemente, especializações de ontologias relacionadas.

A Figura 4.1 ilustra os relacionamentos de especialização entre esses quatro tipos de ontologias, sendo que no topo estão as ontologias genéricas que definem conceitos gerais que embasam todas as outras conceituações e no nível mais baixo estão as ontologias de aplicação, que especializam conceituações para uma determinada classe de aplicações.



**Figura 4.1 - Relacionamentos entre os tipos de ontologias (GUARINO, 1998).**

Uma vez que por meio de ontologias pode-se modelar conhecimento, elas possuem muitas aplicações práticas. Segundo DEVEDZIC (1999), ontologias podem ser utilizadas para integração de bases de conhecimento, uma vez que provêm um esqueleto de conhecimento e uma infra-estrutura independente de uma implementação particular. STAAB et al. (2001), por sua vez, destacam que uma ontologia pode ser usada para derivar visões de conhecimento.

De uma maneira geral, ainda que existam muitas vantagens no uso de ontologias, algumas questões carecem de estudo e esclarecimento, como destaca O'LEARY (apud GUIZZARDI, 2000): (i) como o consenso de muitas partes é dificultado, a definição de uma conceituação compartilhada em uma ontologia é um processo político, no qual pessoas mais significativas em um determinado contexto terão maior influência na conceituação resultante; (ii) ontologias não são necessariamente estacionárias, sendo necessária uma preocupação em evoluí-las constantemente; e (iii) a integração entre ontologias desenvolvidas de forma independente não é uma tarefa trivial, uma vez que podem estar baseadas em conceituações distintas.

Como qualquer artefato de engenharia, uma ontologia deve ser construída seguindo métodos apropriados. Neste trabalho, para desenvolver a ontologia de qualidade de software, foi adotado o método SABiO (*Systematic Approach for Building Ontologies*) (FALBO, 2004), descrito a seguir.

### 4.3 O Método SABiO

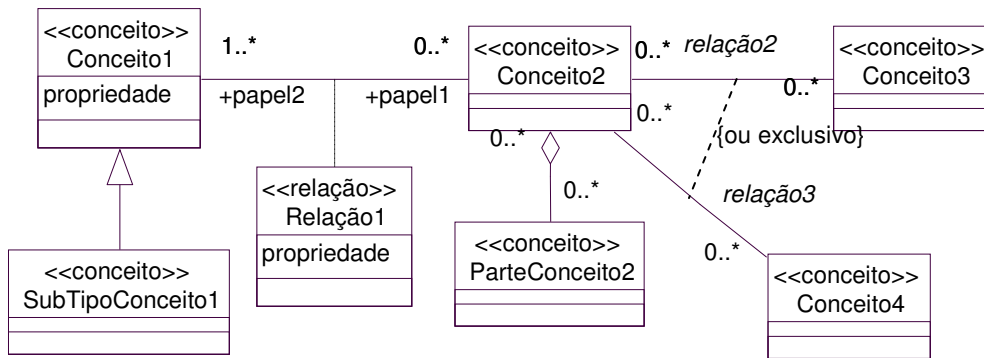
SABiO (*Systematic Approach for Building Ontologies*) (FALBO, 2004) define um processo para construção de ontologias, cujas principais atividades são:

- **Identificação de Propósito e Especificação de Requisitos:** Visa identificar claramente o propósito da ontologia e os usos esperados para ela, i.e., sua competência. Para sua realização, questões de competência devem ser definidas.
- **Captura da Ontologia:** O objetivo é capturar a conceituação do domínio, com base na competência da ontologia. Conceitos, relações, propriedades e axiomas relevantes devem ser identificados e organizados. Um modelo usando uma linguagem gráfica e um dicionário de termos devem ser utilizados para facilitar a comunicação com os especialistas do domínio.
- **Formalização da Ontologia:** Visa representar a conceituação capturada pela ontologia em uma linguagem formal. Neste trabalho, utilizou-se lógica de primeira ordem.
- **Integração com Ontologias Existentes:** Durante a captura ou formalização da ontologia, pode ser necessário integrar a ontologia em questão com outras já existentes, de modo a reutilizar conceituações previamente estabelecidas.
- **Avaliação da Ontologia:** Uma ontologia deve ser avaliada, a fim de verificar se os requisitos estabelecidos na especificação estão sendo satisfeitos, ou seja, dentre outros, a competência da ontologia deve ser verificada.
- **Documentação:** Todo o desenvolvimento da ontologia deve ser documentado.

SABiO advoga, ainda, o uso de uma linguagem de modelagem para facilitar a comunicação dos modelos da ontologia, sugerindo um perfil UML como linguagem gráfica para representação de ontologias (MIAN e FALBO, 2003), adotado neste trabalho. A Figura 4.2 mostra as principais notações desse perfil e alguns dos axiomas definidos para as mesmas.

Para dar uma semântica diferente aos elementos de modelo da UML, foi utilizado o mecanismo de extensão estereótipo. Sendo assim, classes com estereótipo <<conceito>> representam conceitos da ontologia. De maneira análoga, relações que contêm propriedades são representadas como classes associativas com estereótipo <<relação>>.

As relações binárias sem propriedades são definidas como associações nomeadas. As propriedades de conceitos e relações são representadas como atributos das classes estereotipadas. Relações de supertipo e todo-parte são representadas como relações de generalização/especialização e de agregação/composição, respectivamente. Condicionantes entre relações são representados por outro mecanismo de extensão da UML, as restrições entre associações (MIAN, 2003).



#### Axiomas:

##### Todo-Parte

$$(\forall x) \neg \text{parte}(x, x)$$

$$(\forall x, y) \text{ parte}(y, x) \rightarrow \neg \text{parte}(x, y)$$

$$(\forall x, y, z) \text{ parte}(z, y) \wedge \text{parte}(y, x) \rightarrow \text{parte}(z, x)$$

##### Sub-tipo:

$$(\forall x, y, z) (\text{subtipo}(x, y) \wedge \text{subtipo}(y, z) \rightarrow \text{subtipo}(x, z))$$

$$(\forall x, y) (\text{subtipo}(x, y) \rightarrow \text{supertipo}(y, x))$$

##### Condicionante Ou-exclusivo:

$$(\forall a \in C2) ((\exists b) (b \in C3) \wedge R2(a, b)) \rightarrow \neg ((\exists c \in C4) \wedge R3(a, c))$$

$$(\forall a \in C2) ((\exists c) (c \in C4) \wedge R3(a, c)) \rightarrow \neg ((\exists b \in C3) \wedge R2(a, b))$$

**Figura 4.2 – Perfil UML para representar ontologias e axiomas associados.**

Ainda que se esteja utilizando axiomas independentes de domínio para auxiliarem na formalização dos elementos representados pelo perfil da UML utilizado, tais axiomas não são suficientes, uma vez que tratam apenas da estruturação de conceitos e relações. Na formalização de uma ontologia, devem-se também tratar os significados e restrições dos conceitos e relações, bem como as leis de integridade que os regem. Para tanto, podem-se utilizar axiomas dependentes de domínio que podem ser de dois tipos: axiomas de consolidação ou axiomas ontológicos. O primeiro tipo impõe restrições que precisam ser atendidas para que uma relação seja estabelecida

consistentemente. O último pretende representar o conhecimento declarativo que pode derivar novo conhecimento a partir de conhecimento factual representado na ontologia, mas que não é capturado pela estruturação de conceitos e relações da ontologia.

#### ***4.4 A Ontologia de Qualidade de Software***

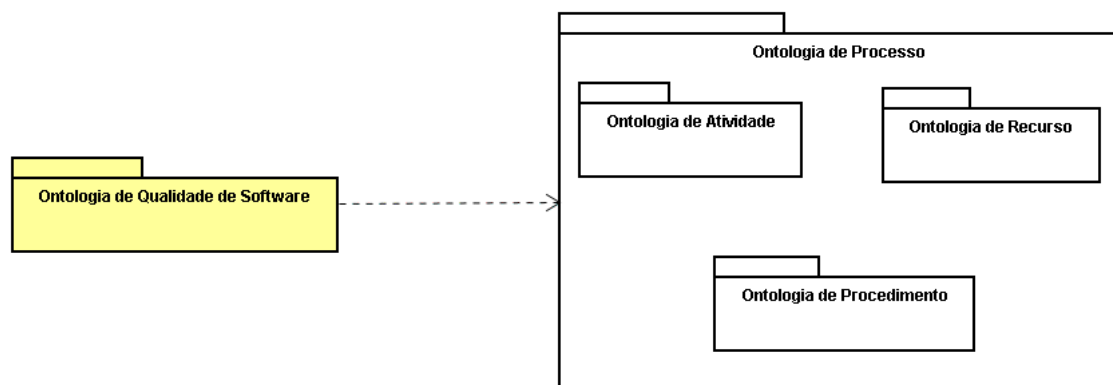
Seguindo o método SABiO, primeiro, identificou-se o propósito da ontologia. Quer se falar de qualidade de software de forma ampla, mas com foco maior para o uso da ontologia como uma especificação conceitual para o desenvolvimento e integração de ferramentas de apoio ao processo de AMP. No que se refere ao domínio, dois aspectos em especial foram levados em conta: (i) ainda que a AMP esteja relacionada a diversos outros processos, o processo de medição ganha destaque e, por isso, aspectos relacionados à medição foram incluídos no escopo; (ii) embora o foco seja avaliação e melhoria de processos, há de se considerar que, para avaliar um processo, é necessário também avaliar os produtos produzidos por ele e, portanto, considera-se também a avaliação de produtos.

Uma vez que processos de software são parte importante do universo de discurso da ontologia de qualidade de software, o seu desenvolvimento foi feito de forma integrada com a ontologia de processo de software proposta em (FALBO e BERTOLLO, 2005). Assim, a conceituação básica relacionada à definição de processos é reutilizada. Além disso, pelo fato de ODE ser totalmente baseado em ontologias, há outras ontologias em sua base ontológica. A figura 4.3 apresenta os relacionamentos da ontologia de qualidade de software com as demais ontologias de ODE.

Vale ressaltar, ainda, que havia uma ontologia de Qualidade de Software em ODE proposta por (DUARTE e FALBO, 2000). Porém, essa ontologia estava fortemente baseada na avaliação da qualidade de produtos, tendo por base a norma ISO/IEC 9126. Além disso, ela também tratava de forma bem simplificada o processo de medição. Desse modo, após uma análise de sua conceituação, percebeu-se que a ontologia não atendia ao escopo deste trabalho, sendo necessária, portanto, uma reformulação considerável da mesma, de modo a contemplar além da avaliação de produtos, também a avaliação de processos de software e a medição. Como as mudanças eram significativas, decidiu-se analisar outros trabalhos na área, tais como (MARTIN e OLSINA, 2003) e (BERTOA et al., 2006), e optou-se por desenvolver uma nova ontologia.

Como o escopo estabelecido agora para a Ontologia de Qualidade de Software é bastante complexo, optou-se por desenvolvê-la como três sub-ontologias integradas, tratando, respectivamente, da estrutura de modelos de qualidade, medição e avaliação. A seguir, cada uma delas é apresentada, procurando mostrar como foi conduzido o seu desenvolvimento, iniciando com a apresentação das questões de competência da ontologia, passando à apresentação dos seus conceitos, modelos e axiomas, e concluindo com a avaliação da ontologia, procurando-se indicar como as questões de competência são respondidas pelos elementos pertencentes à ontologia. Além disso, exemplos de instanciação de cada uma das sub-ontologias segundo as perspectivas de processo e de produto de software são apresentadas.

A integração com a ontologia de processo de software é mostrada nos modelos conceituais apresentados por meio de uma indicação antecedendo o termo. Como a ontologia de processo é decomposta em sub-ontologias de atividade, procedimento e recurso, conforme pode ser observado pela figura 4.3, conceitos reutilizados, tal como Atividade, levam a indicação dessa sub-ontologia, no exemplo Ontologia de Atividade::Atividade.



**Figura 4.3 - Dependência entre as ontologias de Qualidade de Software e de Processo em ODE.**

#### **4.4.1 A Sub-Ontologia de Modelo de Qualidade**

A sub-ontologia de modelo de qualidade trata das características básicas de uma entidade (processos, artefatos, recursos etc) que podem ser medidas, em um primeiro momento, para em seguida serem analisadas e avaliadas. Nas seções que seguem são apresentadas as questões de competência, os conceitos, as relações, os axiomas e as respostas para cada questão de competência identificada para a sub-ontologia de modelo de qualidade.

#### 4.4.1.1 Identificação de Propósito da Sub-Ontologia de Modelo de Qualidade

As questões de competência da sub-ontologia de modelo de qualidade indicam a que se propõe essa ontologia. Basicamente, as questões de competência que devem ser respondidas por essa sub-ontologia são:

QC1. Qual é o tipo de uma entidade mensurável?

QC2. Quais elementos mensuráveis caracterizam um determinado tipo de entidade mensurável?

QC3. Quais elementos mensuráveis podem ser diretamente medidos e quais não podem?

QC4. Como um elemento indiretamente mensurável é decomposto em outros elementos mensuráveis, de modo que possa ser medido a partir da medição destes?

QC5. Quais elementos diretamente mensuráveis devem ser medidos para se poder medir um elemento indiretamente mensurável?

#### 4.4.1.2 Captura e Formalização da Sub-Ontologia de Modelo de Qualidade

A captura da conceituação da sub-ontologia de modelo de qualidade deve ser realizada para atender às questões propostas para ela. Assim, a figura 4.4 apresenta o modelo conceitual dessa sub-ontologia, mostrando seus conceitos e suas relações.

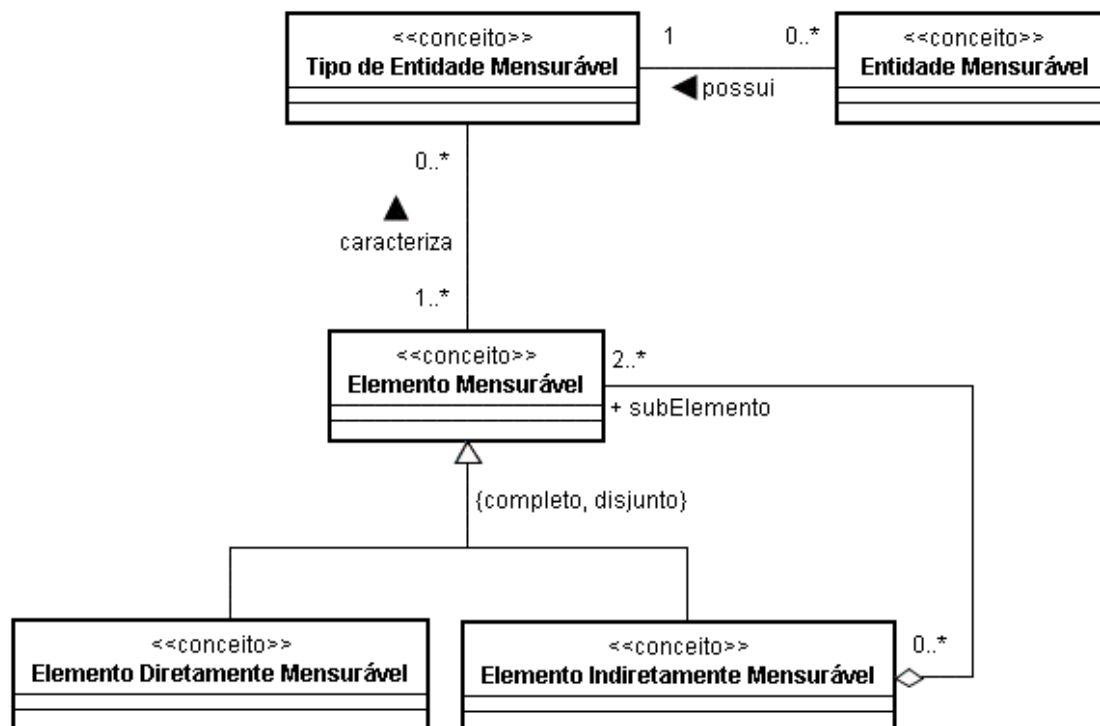


Figura 4.4 – O Modelo da Sub-Ontologia de Modelo de Qualidade.



Um *tipo de entidade mensurável* é uma categoria de entidades mensuráveis que indica quais características (*elementos mensuráveis*) podem ser usadas para medir e avaliar a qualidade de entidades desse tipo. Um *elemento mensurável* pode ser direta ou indiretamente mensurável. Um *elemento diretamente mensurável* (ou atributo) é uma propriedade física ou abstrata de um tipo de entidade mensurável que não necessita de nenhuma outra medida para ser calculada. Já um *elemento indiretamente mensurável* é composto de outros elementos mensuráveis e, sendo assim, necessita de todas as medidas dos elementos que o compõem para ser calculado. É importante destacar que a relação de especialização entre um elemento mensurável e seus subtipos representa um conjunto completo e disjunto, ou seja, todos os elementos mensuráveis são ou direta ou indiretamente mensuráveis (completo), e não há um elemento que seja direta e indiretamente mensurável (disjunto).

Uma *entidade mensurável* é um objeto (instância concreta de um tipo) que pode ser caracterizado mediante a medição dos elementos mensuráveis do seu tipo. São essas entidades que efetivamente estão sendo medidas e avaliadas.

A Tabela 4.1 apresenta a definição dos conceitos da sub-ontologia de modelo de qualidade.

**Tabela 4.1 – Conceitos da Sub-Ontologia de Modelo de Qualidade.**

<b>Conceito</b>	<b>Descrição</b>
<b>Tipo de Entidade Mensurável</b>	Uma categoria de entidades mensuráveis que indica que elementos mensuráveis podem ser usados para medir entidades desse tipo.
<b>Entidade Mensurável</b>	Um objeto que pode ser caracterizado mediante uma medição dos elementos mensuráveis de seu tipo.
<b>Elemento Mensurável</b>	Característica de um tipo de entidade mensurável, por meio da qual a qualidade de suas instâncias pode ser descrita e avaliada.
<b>Elemento Diretamente Mensurável (Atributo)</b>	Uma propriedade, física ou abstrata, de um tipo de entidade mensurável, por meio da qual entidades mensuráveis desse tipo podem ser diretamente medidas, permitindo que sua qualidade seja descrita e avaliada.
<b>Elemento Indiretamente Mensurável</b>	Um conjunto de elementos mensuráveis de um tipo de entidade mensurável, por meio do qual a qualidade de entidades mensuráveis desse tipo pode ser descrita e avaliada.

Uma vez capturados os conceitos da sub-ontologia de modelo de qualidade, ainda há restrições importantes que precisam ser capturadas e descritas na forma de axiomas. Antes, porém, é importante que sejam explicitados os predicados que são usados para formalizar os axiomas dessa sub-ontologia, a saber:

- **caracteriza (em<sub>1</sub>, tem<sub>1</sub>):** indica que o elemento mensurável **em<sub>1</sub>** caracteriza o tipo de entidade mensurável **tem<sub>1</sub>**.
- **subElemento (em<sub>1</sub>, em<sub>2</sub>):** indica que o elemento mensurável **em<sub>1</sub>** é um subelemento mensurável de **em<sub>2</sub>**.
- **elementoDirMensuravel (em<sub>1</sub>):** indica que **em<sub>1</sub>** é um elemento diretamente mensurável.

Definidos os predicados, são apresentados, a seguir, os axiomas da sub-ontologia de modelo de qualidade:

- **A1 – Um elemento indiretamente mensurável *eim<sub>1</sub>* só pode ser decomposto em elementos mensuráveis que caracterizam o mesmo tipo de entidade mensurável *tem<sub>1</sub>*.**

$$(\forall em_1, eim_1, tem_1) subElemento(em_1, eim_1) \rightarrow (caracteriza(em_1, tem_1) \wedge caracteriza(eim_1, tem_1))$$

- **A2 – Se um elemento mensurável *em<sub>1</sub>* é um subelemento de um elemento indiretamente mensurável *eim<sub>1</sub>* e ele é diretamente mensurável, então ele é um subelemento diretamente mensurável de *eim<sub>1</sub>*.**

$$(\forall em_1, eim_1) (subElementoDirMensuravel(em_1, eim_1)) \leftrightarrow (subElemento(em_1, eim_1) \wedge elementoDirMensuravel(em_1))$$

- **A3 – A relação subelemento, assim como quaisquer relações todo-parte, é transitiva.**

$$(\forall em_1, em_2, eim_1) (subElemento(em_1, eim_1)) \wedge (subElemento(em_2, em_1)) \rightarrow subElemento(em_2, eim_1)$$

#### 4.4.1.3 Avaliação da Sub-Ontologia de Modelo de Qualidade

A última etapa do processo de desenvolvimento de uma ontologia é a sua avaliação. Particularmente, está-se interessado em avaliar se a ontologia responde às questões de competência colocadas. Avaliando modelos e axiomas, pode-se perceber que algumas questões são diretamente respondidas por uma relação do modelo, tal como QC1, que é respondida pela relação *possui* entre os conceitos *Entidade*

*Mensurável* e *Tipo de Entidade Mensurável*. Outras dependem de várias relações e alguns axiomas. Por exemplo, para responder a QC5, são necessárias a relação de *sub-tipo de* entre *Elemento Mensurável* e seus subtipos, a relação *agrega* entre *Elemento Indiretamente Mensurável* e *Elemento Mensurável* e os axiomas (A2) e (A3). A Tabela 4.2 mostra a avaliação completa da sub-ontologia de modelo de qualidade.

**Tabela 4.2 – Avaliação da Sub-Ontologia de Modelo de Qualidade.**

<b>Questão de Competência</b>	<b>Conceito A</b>	<b>Relação</b>	<b>Conceito B</b>	<b>Axiomas</b>
QC1	Entidade Mensurável	possui	Tipo de Entidade Mensurável	-
QC2	Elemento Mensurável	caracteriza	Tipo de Entidade Mensurável	-
QC3	Elemento Mensurável	é super tipo de	Elemento Diretamente Mensurável e Elemento Indiretamente Mensurável	-
QC4	Elemento Indiretamente Mensurável	agrega	Elemento Mensurável	A1
QC5	Elemento Mensurável	é super tipo de	Elemento Diretamente Mensurável e Elemento Indiretamente Mensurável	A2 e A3
	Elemento Indiretamente Mensurável	agrega	Elemento Mensurável	

#### **4.4.1.4 Exemplos de Instanciação da Sub-Ontologia de Modelo de Qualidade**

Com o intuito de ilustrar a utilidade da ontologia tanto para falar sobre processo quanto para tratar de produto de software, são apresentados na Tabela 4.3 dois exemplos de instanciação da sub-ontologia de modelo da qualidade, na qual a primeira coluna

representa um exemplo relacionado a processo de software enquanto a segunda apresenta um exemplo relacionado a produto de software. Vale notar que o exemplo instanciado permite, inclusive, mostrar a possibilidade de instanciar um elemento diretamente mensurável para processo e outro indiretamente mensurável para produto. Esses mesmos exemplos são usados na instanciamento das outras sub-ontologias, mostrando, por exemplo, como se dá a medição de um elemento diretamente mensurável e outro indiretamente mensurável.

**Tabela 4.3 – Instanciamento da Sub-Ontologia de Modelo de Qualidade.**

<b>Tipo de Entidade Mensurável</b>	<b>Processos de uma Unidade Organizacional</b>	<b>Especificação de Requisitos</b>
<b>Entidade Mensurável</b>	Processo de Gerência de Projetos da Divisão de Desenvolvimento de Sistemas da Empresa X	Especificação de Requisitos do Projeto 1
<b>Elemento Mensurável</b>	Resultado Esperado do MPS.BR GPR 3 em um Projeto	Adequação do Uso do Método na Elaboração da Especificação de Requisitos
<b>Elemento Diretamente Mensurável</b>	Resultado Esperado do MPS.BR GPR 3 em um Projeto	-
<b>Elemento Indiretamente Mensurável</b>	-	Adequação do Uso do Método na Elaboração da Especificação de Requisitos

#### **4.4.2 A Sub-Ontologia de Medição**

A sub-ontologia de medição trata, como o próprio nome sugere, da medição das características básicas de uma entidade. Seu objetivo é medir, por meio da medição de elementos mensuráveis do tipo de uma entidade, características dessa entidade.

##### **4.4.2.1 Identificação de Propósito da Sub-Ontologia de Medição**

A sub-ontologia de medição deve responder as seguintes questões de competência:

QC1. Que entidade mensurável está sendo medida em uma medição?

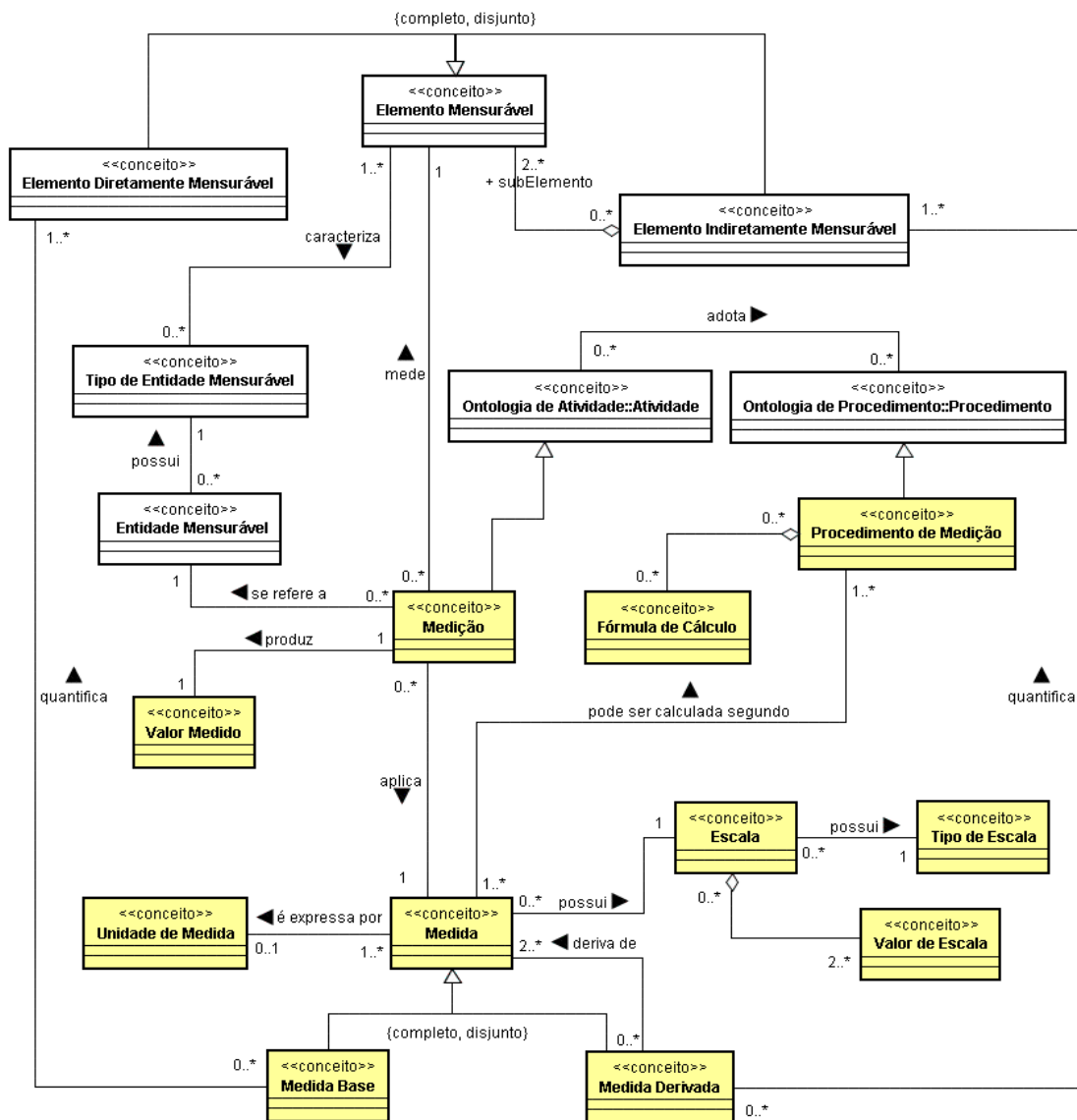
QC2. Qual é o tipo da entidade mensurável que está sendo medida em uma medição?

QC3. Que elementos mensuráveis de um tipo de entidade mensurável podem ser medidos para se avaliar a qualidade de uma entidade mensurável associada a esse tipo?

- QC4. Qual é o elemento mensurável da entidade mensurável que está sendo medido em uma medição?
- QC5. Que medidas podem ser utilizadas para quantificar um elemento mensurável?
- QC6. Que medida está sendo aplicada em uma medição de um elemento mensurável?
- QC7. Quanto à dependência de uma medida em relação a outras, qual a natureza de uma medida?
- QC8. Que medidas base podem ser utilizadas para quantificar um elemento diretamente mensurável?
- QC9. Que medidas derivadas podem ser utilizadas para quantificar um elemento indiretamente mensurável?
- QC10. Que medidas precisam ser medidas para que uma medida derivada possa ser computada?
- QC11. Que medidas base precisam ser medidas para que uma medida derivada seja computada?
- QC12. Qual a unidade de medida que expressa uma determinada medida?
- QC13. Qual é a escala de uma medida?
- QC14. Qual é o tipo de uma escala?
- QC15. Quais são os valores de uma escala?
- QC16. Qual o valor medido em uma medição?
- QC17. Que procedimentos de medição podem ser usados para calcular uma medida?
- QC18. Quais as fórmulas de cálculo envolvidas em um procedimento de medição?
- QC19. Que procedimentos de medição estão sendo adotados em uma medição?

#### **4.4.2.2 Captura e Formalização da Sub-Ontologia de Medição**

Visando atender às questões de competência propostas, foram identificados conceitos, relações e axiomas que visam descrever o domínio de medição. A Figura 4.5 mostra o modelo conceitual da sub-ontologia de medição, destacando em amarelo seus conceitos.



**Figura 4.5 – O Modelo da Sub-Ontologia de Medição.**

A *medição* é uma atividade que tem por objetivo medir uma característica (elemento mensurável) de uma entidade, aplicando uma *medida* para obter um *valor*. Uma medida pode ser *base*, caso ela seja funcionalmente independente de outras medidas, ou *derivada*, caso seja definida como função de duas ou mais medidas (ISO/IEC, 2002). Uma medida base quantifica elementos diretamente mensuráveis, enquanto uma medida derivada quantifica elementos indiretamente mensuráveis.

Uma *medida* possui uma *escala* relacionada e é expressa segundo uma *unidade de medida*. Além disso, uma medida é calculada segundo um *procedimento de medição* que pode conter uma *fórmula de cálculo*. Ao se aplicar uma medida em uma medição, o procedimento de medição adotado deve ser apropriado para a medida em questão.

A Tabela 4.4 apresenta a definição em linguagem natural dos conceitos da sub-ontologia de medição.

**Tabela 4.4 – Conceitos da Sub-Ontologia de Medição.**

<b>Conceito</b>	<b>Descrição</b>
<b>Medição</b>	Atividade que visa medir um elemento mensurável de uma entidade mensurável, obtendo-se um valor medido pela aplicação de uma medida.
<b>Medida</b>	Variável para a qual é atribuído um valor durante uma medição.
<b>Medida Base</b>	Medida funcionalmente independente de outras medidas, usada para medir elementos diretamente mensuráveis (atributos).
<b>Medida Derivada</b>	Medida que é definida como uma função de duas ou mais medidas e é usada para quantificar um elemento indiretamente mensurável.
<b>Valor Medido</b>	Resultado atribuído a uma medida em uma medição.
<b>Unidade de Medida</b>	Quantidade particular, definida e adotada por convenção, com as quais outras quantidades do mesmo tipo são comparadas a fim de expressar sua magnitude relativa a essa outra quantidade.
<b>Procedimento de Medição</b>	Uma sequência lógica de operações usadas para se calcular uma medida, a ser adotada na realização de medições de elementos mensuráveis.
<b>Fórmula de Cálculo</b>	Fórmula utilizada por um procedimento de medição.
<b>Escala</b>	Um conjunto de valores (discreto ou contínuo) ou categorias para o qual uma medida é mapeada.
<b>Tipo de Escala</b>	Indica a natureza da relação entre os valores de uma escala.
<b>Valor de Escala</b>	Número ou categoria que compõe o conjunto de valores de uma escala.

Além dos conceitos específicos da sub-ontologia de medição, há também conceitos reutilizados da ontologia de processo de software. Assim, a Tabela 4.5 apresenta as definições desses conceitos (Procedimento e Atividade), bem como do conceito de Artefato, reutilizado na sub-ontologia de Avaliação.

**Tabela 4.5 – Conceitos Reutilizados da Ontologia de Processo de Software.**

Conceito	Descrição	Ontologia de Origem
<b>Procedimento</b>	Conduta bem estabelecida e ordenada para a realização de atividades.	Ontologia de Procedimento
<b>Atividade</b>	Uma porção do trabalho a ser realizado, que pode utilizar artefatos como entrada e produzir artefatos como saída.	Ontologia de Atividade
<b>Artefato</b>	Um insumo para, ou um produto de, uma atividade, no sentido de ser um objeto de transformação da atividade.	Ontologia de Atividade

Vale destacar que a ontologia de processo de software foi recentemente analisada e re-projetada para deixar explícitas algumas distinções ontológicas que estavam apenas implícitas, mapeando seus conceitos nos conceitos da Ontologia Fundamental Unificada (*Unified Foundational Ontology* - UFO) (GUIZZARDI et al., 2008). Em (GUIZZARDI et al., 2008), faz-se a distinção entre tipos de atividade e ocorrências de atividades e entre tipos de artefatos e artefatos concretos. Ainda que as ontologias desenvolvidas neste trabalho não tenham sido construídas tomando por base a UFO, assume-se que os conceitos de Atividade e Artefato se referem a ocorrências e não tipos.

Uma vez capturados os conceitos e relações da sub-ontologia de medição, apresentam-se as restrições existentes nessa sub-ontologia, descritas na forma de axiomas. Contudo, antes de se apresentarem os axiomas, é necessário que sejam descritos os predicados que são utilizados em sua formalização, a saber:

- **mede (med<sub>1</sub>, em<sub>1</sub>):** indica que a medição **med<sub>1</sub>** mede o elemento mensurável **em<sub>1</sub>**.
- **aplica (med<sub>1</sub>, m<sub>1</sub>):** indica que a medição **med<sub>1</sub>** aplica a medida **m<sub>1</sub>**.
- **quantifica (m<sub>1</sub>, em<sub>1</sub>):** indica que a medida **m<sub>1</sub>** quantifica o elemento mensurável **em<sub>1</sub>**.
- **caracteriza (em<sub>1</sub>, tem<sub>1</sub>):** indica que o elemento mensurável **em<sub>1</sub>** caracteriza o tipo de entidade mensurável **tem<sub>1</sub>**.



- **seRefereA (med<sub>1</sub>, entm<sub>1</sub>):** indica que a medição **med<sub>1</sub>** se refere à entidade mensurável **entm<sub>1</sub>**.
- **possui (entm<sub>1</sub>, tem<sub>1</sub>):** indica que a entidade mensurável **entm<sub>1</sub>** possui o tipo de entidade mensurável **tem<sub>1</sub>**.
- **adota (a<sub>1</sub>, p<sub>1</sub>):** indica que a atividade **a<sub>1</sub>** adota o procedimento **p<sub>1</sub>**.
- **podeSerCalculadaSegundo (m<sub>1</sub>, pm<sub>1</sub>):** indica que a medida **m<sub>1</sub>** pode ser calculada segundo o procedimento de medição **pm<sub>1</sub>**.
- **derivaDe (md<sub>1</sub>, m<sub>1</sub>):** indica que a medida derivada **md<sub>1</sub>** deriva da medida **m<sub>1</sub>**.
- **medidaBase (m<sub>1</sub>):** indica que **m<sub>1</sub>** é uma medida base.
- **medidaDerivada (m<sub>1</sub>):** indica que **m<sub>1</sub>** é uma medida derivada.
- **subElemento (em<sub>1</sub>, em<sub>2</sub>):** indica que o elemento mensurável **em<sub>1</sub>** é um subelemento de **em<sub>2</sub>**.
- **elementoDirMensuravel (em<sub>1</sub>):** indica que **em<sub>1</sub>** é um elemento diretamente mensurável.
- **elementoIndMensuravel (em<sub>1</sub>):** indica que **em<sub>1</sub>** é um elemento indiretamente mensurável.

Definidos os predicados, apresentam-se a seguir os axiomas da sub-ontologia de medição:

- **A1 – Uma medida *m<sub>1</sub>* só pode ser aplicada em uma medição *med<sub>1</sub>*, se a medição *med<sub>1</sub>* estiver medindo um elemento mensurável *em<sub>1</sub>* quantificável pela medida *m<sub>1</sub>*.**

$$(\forall em_1, m_1, med_1) \text{ aplica}(med_1, m_1) \rightarrow (mede(med_1, em_1) \wedge quantifica(m_1, em_1))$$

- **A2 – Uma medição *med<sub>1</sub>* que se refere a uma entidade mensurável *entm<sub>1</sub>* só pode medir elementos mensuráveis que caracterizam o tipo dessa entidade.**

$$(\forall em_1, tem_1, entm_1, med_1) ( seRefereA(med_1, entm_1) \wedge mede(med_1, em_1) ) \rightarrow ( possui(entm_1, tem_1) \wedge caracteriza(em_1, tem_1) )$$

- **A3 – Uma atividade de medição *med<sub>1</sub>* que aplica uma medida *m<sub>1</sub>* só pode adotar um procedimento de medição *pm<sub>1</sub>* se a medida *m<sub>1</sub>* puder ser calculada segundo esse procedimento de medição.**

$$(\forall m_1, med_1, pm_1) ( aplica(med_1, m_1) \wedge adota(med_1, pm_1) ) \rightarrow podeSerCalculadaSegundo(m_1, pm_1)$$

- **A4 – Se uma medição  $med_1$  aplica uma medida  $m_1$  para medir um elemento diretamente mensurável  $em_1$ , então a medida  $m_1$  deve ser uma medida base que quantifique o elemento diretamente mensurável  $em_1$ .**

$$(\forall med_1, m_1, em_1) (mede(med_1, em_1) \wedge elemetoDirMensuravel(em_1) \wedge aplica(med_1, m_1)) \rightarrow (quantifica(m_1, em_1) \wedge medidaBase(m_1))$$

- **A5 – Se uma medição  $med_1$  aplica uma medida  $m_1$  para medir um elemento indiretamente mensurável  $em_1$ , então a medida  $m_1$  deve ser uma medida derivada que quantifique esse elemento.**

$$(\forall med_1, m_1, em_1) (mede(med_1, em_1) \wedge elemetoIndMensuravel(em_1) \wedge aplica(med_1, m_1)) \rightarrow (quantifica(m_1, em_1) \wedge medidaDerivada(m_1))$$

- **A6 – Se uma medida derivada  $md_1$  quantifica um elemento indiretamente mensurável  $eim_1$  e ela deriva de uma medida  $m_2$ , então existe um elemento mensurável  $em_2$  que é quantificado pela medida  $m_2$  e é um sub-elemento de  $eim_1$ .**

$$(\forall md_1, m_2, eim_1, em_2) (quantifica(md_1, eim_1) \wedge derivaDe(md_1, m_2)) \rightarrow (\exists em_2 (quantifica(m_2, em_2) \wedge subElemento(em_2, eim_1)))$$

#### 4.4.2.3 Avaliação da Sub-Ontologia de Medição

A sub-ontologia de medição é mais ampla e complexa que a sub-ontologia de modelo de qualidade. Isso pode ser percebido tanto pelo número de conceitos que ela possui como também pela quantidade de questões de competência que devem ser avaliadas. Novamente, para avaliar a sub-ontologia é necessário que sejam respondidas as suas questões de competência. Logo, analisando os modelos e axiomas, percebe-se que há questões cujas respostas são obtidas por um único relacionamento, como é o caso de QC1, que é respondida pela relação *se refere a* entre os conceitos *Medição* e *Entidade Mensurável*. Contudo, há questões mais complexas que envolvem vários relacionamentos e axiomas para serem respondidas, como é o caso de QC6, onde são necessárias seis relações e três axiomas para respondê-la. As relações para responder QC6 são: (i) *mede* entre *Medição* e *Elemento Mensurável*; (ii) *aplica* entre *Medição* e *Medida*; (iii) *é super tipo de* entre *Medida* e seus subtipos *Medida Base* e *Medida Derivada*; (iv) *é super tipo de* entre *Elemento Mensurável* e seus subtipos *Elemento Diretamente Mensurável* e *Elemento Indiretamente Mensurável*; (v) *quantifica* entre

*Medida Base e Elemento Diretamente Mensurável*; e, por fim, (vi) *quantifica* entre *Medida Derivada e Elemento Indiretamente Mensurável*. Além das relações apresentadas, também são necessários os axiomas (A1), (A4) e (A5).

A Tabela 4.6 mostra a avaliação completa da sub-ontologia de medição.

**Tabela 4.6 – Avaliação da Sub-Ontologia de Medição.**

<b>Questão de Competência</b>	<b>Conceito A</b>	<b>Relação</b>	<b>Conceito B</b>	<b>Axiomas</b>
QC1	Medição	se refere a	Entidade Mensurável	-
QC2	Medição	se refere a	Entidade Mensurável	-
	Entidade Mensurável	possui	Tipo de Entidade Mensurável	
QC3	Medição	se refere a	Entidade Mensurável	A2
	Entidade Mensurável	possui	Tipo de Entidade Mensurável	
	Medição	mede	Elemento Mensurável	
	Elemento Mensurável	caracteriza	Tipo de Entidade Mensurável	
QC4	Medição	mede	Elemento Mensurável	-
QC5	Medida Base	quantifica	Elemento Diretamente Mensurável	A6
	Medida Derivada	quantifica	Elemento Indiretamente Mensurável	
	Medida Derivada	deriva de	Medida	
	Elemento Mensurável	é super tipo de	Elemento Diretamente Mensurável e Elemento Indiretamente Mensurável	
	Elemento Indiretamente Mensurável	agrega	Elemento Mensurável	

**Tabela 4.6 – Avaliação da Sub-Ontologia de Medição (continuação).**

<b>Questão de Competência</b>	<b>Conceito A</b>	<b>Relação</b>	<b>Conceito B</b>	<b>Axiomas</b>
QC6	Medição	mede	Elemento Mensurável	A1, A4 e A5
	Medição	aplica	Medida	
	Medida	é super tipo de	Medida Base e Medida Derivada	
	Elemento Mensurável	é super tipo de	Elemento Diretamente Mensurável e Elemento Indiretamente Mensurável	
	Medida Base	quantifica	Elemento Diretamente Mensurável	
	Medida Derivada	quantifica	Elemento Indiretamente Mensurável	
QC7	Medida	é super tipo de	Medida Base e Medida Derivada	-
QC8	Medida Base	quantifica	Elemento Diretamente Mensurável	
QC9	Medida Derivada	quantifica	Elemento Indiretamente Mensurável	
QC10	Medida Derivada	deriva de	Medida	A6
	Elemento Mensurável	é super tipo de	Elemento Indiretamente Mensurável e Elemento Diretamente Mensurável	
	Medida Base	quantifica	Elemento Diretamente Mensurável	
	Medida Derivada	quantifica	Elemento Indiretamente Mensurável	
	Elemento Indiretamente Mensurável	agrega	Elemento Mensurável	

**Tabela 4.6 – Avaliação da Sub-Ontologia de Medição (continuação).**

<b>Questão de Competência</b>	<b>Conceito A</b>	<b>Relação</b>	<b>Conceito B</b>	<b>Axiomas</b>
QC11	Medida Derivada	deriva de	Medida	A6
	Elemento Mensurável	é super tipo de	Elemento Indiretamente Mensurável e Elemento Diretamente Mensurável	
	Medida Base	quantifica	Elemento Diretamente Mensurável	
	Medida Derivada	quantifica	Elemento Indiretamente Mensurável	
	Medida	é super tipo de	Medida Base e Medida Derivada	
	Elemento Indiretamente Mensurável	agrega	Elemento Mensurável	
QC12	Medida	é expressa por	Unidade de Medida	-
QC13	Medida	possui	Escala	-
QC14	Escala	possui	Tipo de Escala	-
QC15	Escala	agrega	Valor de Escala	-
QC16	Medição	produz	Valor Medido	-
QC17	Medida	pode ser calculada segundo	Procedimento de Medição	-
QC18	Procedimento de Medição	agrega	Fórmula de Cálculo	-
QC19	Atividade	adota	Procedimento	A3
	Medição	é sub tipo de	Atividade	
	Procedimento de Medição	é sub tipo de	Procedimento	
	Medição	aplica	Medida	
	Medida	pode ser calculada segundo	Procedimento de Medição	

#### 4.4.2.4 Exemplos de Instanciação da Sub-Ontologia de Medição

A Tabela 4.7 apresenta a instanciação da sub-ontologia de medição seguindo a mesma estrutura definida para a sub-ontologia de modelo de qualidade. Assim, são apresentados dois exemplos de instanciação da sub-ontologia de medição, sendo que a primeira coluna representa um exemplo relacionado a processo de software e a segunda um exemplo relacionado a produto de software. Novamente, vale notar que nos exemplos instanciados foram consideradas uma medida base referente à medição de um elemento diretamente mensurável, no caso de processo, e uma medida derivada referente à medição de um elemento indiretamente mensurável, no caso de produto.

**Tabela 4.7 – Instanciação da Sub-Ontologia de Medição.**

<b>Medida</b>	Grau de Implementação de Resultado Esperado do Processo em Projeto	Correção no Uso do Método
<b>Medida Base</b>	Grau de Implementação de Resultado Esperado do Processo em Projeto	-
<b>Medida Derivada</b>	-	Correção no Uso do Método
<b>Escala</b>	Escala para caracterização do grau de implementação de um resultado esperado do processo e de um resultado esperado de atributo do processo	Escala para caracterização da adequação do uso do método na elaboração de uma especificação de requisitos
<b>Tipo de Escala</b>	Ordinal	Absoluta
<b>Valores de Escala</b>	[T, L, P, N, NA, F]	$[0 \leq X \leq 1]$

**Tabela 4.7 – Instanciação da Sub-Ontologia de Medição (continuação).**

<b>Procedimento de Medição</b>	<p>Considerar <b>T</b> se:</p> <ul style="list-style-type: none"> <li>(i) O indicador direto está presente e é julgado adequado;</li> <li>(ii) Existe pelo menos um indicador indireto e/ou afirmação confirmando a implementação;</li> <li>(iii) Não foi notado nenhum ponto fraco substancial;</li> </ul> <p>Considerar <b>L</b> se:</p> <ul style="list-style-type: none"> <li>(i) O indicador direto está presente e é julgado adequado;</li> <li>(ii) Existe pelo menos um indicador indireto e/ou afirmação confirmando a implementação;</li> <li>(iii) Foi notado um ou mais pontos fracos substanciais.</li> </ul> <p>Considerar <b>P</b> se:</p> <ul style="list-style-type: none"> <li>(i) O indicador direto não está presente ou é julgado inadequado;</li> <li>(ii) Artefatos/afirmações sugerem que alguns aspectos do resultado esperado estão implementados;</li> <li>(iii) Pontos fracos foram documentados.</li> </ul> <p>Considerar <b>N</b> se:</p> <ul style="list-style-type: none"> <li>(i) Qualquer situação diferente das acima.</li> </ul> <p>Considerar <b>NA</b> se:</p> <ul style="list-style-type: none"> <li>(i) O projeto não está na fase de desenvolvimento que permite atender ao resultado ou não faz parte do escopo do projeto atender o resultado.</li> </ul> <p>Considerar <b>F</b> se:</p> <ul style="list-style-type: none"> <li>(i) O resultado esperado está fora do escopo de avaliação, conforme documentado no plano de avaliação.</li> </ul>	<p>Este critério pode ser avaliado da seguinte maneira:</p> <ul style="list-style-type: none"> <li>(i) Calcule o valor medido para a medida base Correção da Notação;</li> <li>(ii) Calcule o valor medido para a medida base Correção Sintática;</li> <li>(iii) Use a fórmula de cálculo abaixo para calcular a correção no uso do método.</li> </ul>
<b>Fórmula de Cálculo</b>	-	$X = (0,7)*A + (0,3)*B,$ <p>onde  A = valor medido da medida base Correção Sintática;  B = valor medido da medida base Correção da Notação.</p>
<b>Medição</b>	Caracterizar o grau de implementação do resultado esperado de processo MPS.BR GPR 3 no Projeto P1	Avaliar a correção do uso do método da Especificação de Requisitos do Projeto 1
<b>Valor Medido</b>	L	0,7

### **4.4.3 A Sub-Ontologia de Avaliação**

A sub-ontologia de avaliação visa retratar o universo de avaliações da qualidade de uma entidade por meio da análise de medições, incluindo a indicação de ações a serem tomadas, propostas a partir de observações da avaliação dessa entidade.

#### **4.4.3.1 Identificação de Propósito da Sub-Ontologia de Avaliação**

As questões de competência que devem ser respondidas pela sub-ontologia de avaliação são:

- QC1. Quais são os modelos de análise de medição definidos para uma medida?
- QC2. Quais são os critérios de decisão de um modelo de análise de medição?
- QC3. Que premissas compõem um critério de decisão?
- QC4. Qual a conclusão de um critério de decisão?
- QC5. Que modelo de análise de medição é adotado em uma análise de medição?
- QC6. Qual é o valor medido analisado durante uma atividade de análise de medição?
- QC7. De que atividade de medição depende uma atividade de análise de medição?
- QC8. Qual medida está sendo analisada em uma atividade de análise de medição?
- QC9. Qual é a conclusão de uma atividade de análise de medição?
- QC10. De que atividades de análise de medição depende uma atividade de avaliação da qualidade?
- QC11. Quais são os tipos de observação de avaliação?
- QC12. Que observações de avaliação foram produzidas em uma avaliação?
- QC13. Quais são os tipos de ação de qualidade?
- QC14. Quais as ações de qualidade geradas a partir de uma observação de avaliação?

#### **4.4.3.2 Captura e Formalização da Sub-Ontologia de Avaliação**

Para responder às questões de competência propostas pela sub-ontologia de avaliação, foram identificados conceitos, relações e axiomas. A Figura 4.6 mostra o modelo conceitual da sub-ontologia de avaliação, destacando em amarelo seus conceitos.





qualidade. Um ponto fraco é uma característica considerada inadequada ou que não atende aos requisitos da entidade. Uma oportunidade de melhoria é uma característica que pode ser melhorada, mas que atende aos requisitos mínimos colocados para a entidade. Por fim, uma não conformidade é um não atendimento de um requisito de qualidade (ABNT, 2000).

A partir de observações realizadas numa avaliação da qualidade de uma entidade, *ações corretivas*, *preventivas* ou *de melhoria* podem ser apontadas. Uma ação corretiva visa eliminar a causa de uma não conformidade identificada ou outra situação indesejável de uma entidade (ABNT, 2000). Uma ação preventiva visa eliminar a causa de uma não conformidade potencial ou outra situação potencialmente indesejável (ABNT, 2000). Uma ação de melhoria visa aumentar a capacidade de satisfazer os requisitos de qualidade já existentes de uma entidade mensurável (ABNT, 2000).

A Tabela 4.8 apresenta a definição em linguagem natural dos conceitos da sub-ontologia de avaliação.

**Tabela 4.8 – Conceitos da Sub-Ontologia de Avaliação.**

<b>Conceito</b>	<b>Descrição</b>
<b>Modelo de Análise de Medição</b>	Um procedimento descrevendo critérios de decisão para se analisar valores medidos obtidos em medições de uma ou mais medidas.
<b>Análise de Medição</b>	Atividade que visa analisar valores medidos de medidas, adotando um modelo de análise de medição para atribuir uma conclusão que indique, de algum modo, uma característica da entidade sendo mensurada.
<b>Critério de Decisão</b>	Sentença que estabelece uma conclusão a partir de um conjunto de premissas, descrevendo o nível de confiança de um resultado.
<b>Premissa</b>	Fato ou princípio, de um critério de decisão, que serve de base para uma conclusão desse critério.
<b>Conclusão</b>	Uma inferência (ou consequência lógica) decorrente de um conjunto de premissas.
<b>Avaliação de Qualidade</b>	Atividade que visa avaliar sistematicamente quanto uma entidade é capaz de atender os requisitos especificados.

**Tabela 4.8 – Conceitos da Sub-Ontologia de Avaliação (continuação).**

<b>Conceito</b>	<b>Descrição</b>
<b>Observação de Qualidade</b>	Evidências objetivas documentadas produzidas em uma atividade de avaliação da qualidade.
<b>Ponto Forte</b>	Uma característica excepcionalmente boa de uma entidade mensurável, observada em uma avaliação de qualidade.
<b>Ponto Fraco</b>	Uma característica considerada inadequada ou que não atende aos requisitos de uma entidade mensurável, observada em uma avaliação de qualidade.
<b>Oportunidade de Melhoria</b>	Uma característica que pode ser melhorada, mas que atende aos requisitos mínimos de uma entidade mensurável, observada em uma avaliação de qualidade.
<b>Não Conformidade</b>	Não atendimento de um requisito de qualidade.
<b>Ação de Qualidade</b>	Ação apontada a partir de observações realizadas numa avaliação de qualidade de uma entidade mensurável.
<b>Ação Preventiva</b>	Ação de qualidade definida para eliminar a causa de uma não conformidade potencial ou outra situação potencialmente indesejável de uma entidade mensurável.
<b>Ação Corretiva</b>	Ação de qualidade definida para eliminar a causa de uma não conformidade identificada ou outra situação indesejável de uma entidade mensurável.
<b>Ação de Melhoria</b>	Ação de qualidade que visa o aumento da capacidade de satisfazer os requisitos de qualidade já existentes de uma entidade mensurável.

Assim como nas demais sub-ontologias, o modelo apresentando na Figura 4.6 não é capaz de capturar toda a conceituação do domínio, sendo necessário descrever restrições na forma de axiomas. Antes, porém, são apresentados os predicados que apóiam a definição dos axiomas. São eles:

- **adota (a1, p1):** indica que a atividade **a1** adota o procedimento **p1**.
- **produto (art1, a1):** indica que o artefato **art1** é um produto da atividade **a1**.
- **analiseMedicao (a1):** indica que a atividade **a1** é uma atividade de análise de medição.
- **avaliacaoQualidade (a1):** indica que a atividade **a1** é uma atividade de avaliação da qualidade.

- **observacaoAvaliacao (art<sub>1</sub>):** indica que o artefato **art<sub>1</sub>** é uma observação de avaliação.
- **modeloAnaliseMedicao (p<sub>1</sub>):** indica que o procedimento **p<sub>1</sub>** é um modelo de análise de medição.
- **dependeDe (a<sub>1</sub>, a<sub>2</sub>):** indica que a atividade **a<sub>1</sub>** depende da atividade **a<sub>2</sub>**.
- **conclusaoDeCritério (c<sub>1</sub>, cd<sub>1</sub>):** indica que a conclusão **c<sub>1</sub>** é a conclusão do critério de decisão **cd<sub>1</sub>**.
- **premissaDeCritério (p<sub>1</sub>, cd<sub>1</sub>):** indica que a premissa **p<sub>1</sub>** é uma das premissas do critério de decisão **cd<sub>1</sub>**.
- **ehDefinidoPor (mam<sub>1</sub>, cd<sub>1</sub>):** indica que o modelo de análise de medição **mam<sub>1</sub>** é definido pelo critério de decisão **cd<sub>1</sub>**.
- **atribui (a<sub>1</sub>, c<sub>1</sub>):** indica que a atividade de análise de medição **a<sub>1</sub>** atribui uma conclusão **c<sub>1</sub>**.
- **analisa (a<sub>1</sub>, v<sub>1</sub>):** indica que a atividade de análise de medição **a<sub>1</sub>** analisa o valor medido **v<sub>1</sub>**.
- **aplica (med<sub>1</sub>, m<sub>1</sub>):** indica que a atividade de medição **med<sub>1</sub>** aplica a medida **m<sub>1</sub>**.
- **ehDefinidoPara (mam<sub>1</sub>, m<sub>1</sub>):** indica que o modelo de análise de medição **mam<sub>1</sub>** é definido para a medida **m<sub>1</sub>**.
- **produz (med<sub>1</sub>, v<sub>1</sub>):** indica que a atividade de medição **med<sub>1</sub>** produz o valor medido **v<sub>1</sub>**.
- **gera (o<sub>1</sub>, aq<sub>1</sub>):** indica que a observação de medição **o<sub>1</sub>** gera a ação de qualidade **aq<sub>1</sub>**.

Uma vez definidos os predicados, a seguir são apresentados os axiomas da sub-ontologia de avaliação:

- **A1 – Uma análise de medição  $a_I$  analisa a medida  $m_I$  se e somente se a análise de medição  $a_I$  analisa um valor medido  $v_I$  e esse valor é produzido por uma medição  $med_I$  que aplica a medida  $m_I$ .**

$$(\forall a_I, v_I, med_I, m_I) ( analisa(a_I, v_I) \wedge produz(med_I, v_I) \wedge aplica (med_I, m_I) ) \leftrightarrow analisaMedida(a_I, m_I)$$

- **A2 – Se uma atividade de análise de medição  $a_1$  analisa uma medida  $m_1$ , então um procedimento  $p_1$  dentre os procedimentos adotados por  $a_1$  deve ser um modelo de análise de medição definido para a medida  $m_1$ .**

$$(\forall a_1, m_1) \text{ analisaMedida}(a_1, m_1) \rightarrow (\exists p_1) ( \text{adota}(a_1, p_1) \wedge \text{modeloAnaliseMedicao}(p_1) \wedge \text{ehDefinidoPara}(p_1, m_1) )$$

- **A3 – Uma atividade de análise de medição só pode adotar um único modelo de análise de medição.**

$$(\forall a_1, p_1, p_2) ( \text{analiseMedicao}(a_1) \wedge \text{adota}(a_1, p_1) \wedge \text{adota}(a_1, p_2) \wedge \text{modeloAnaliseMedicao}(p_1) ) \rightarrow \neg \text{modeloAnaliseMedicao}(p_2)$$

- **A4 – Uma análise de medição  $a_1$  só pode atribuir uma conclusão  $c_1$ , se  $c_1$  faz parte de um critério de decisão  $cd_1$  definido para o modelo de análise de medição  $mam_1$  adotado por  $a_1$ .**

$$(\forall a_1, m_1, c_1, cd_1) \text{ atribui}(a_1, c_1) \rightarrow ( \text{conclusaoDeCritério}(c_1, cd_1) \wedge \text{ehDefinidoPor}(mam_1, cd_1) \wedge \text{adota}(a_1, mam_1) )$$

- **A5 – Se uma atividade é uma avaliação da qualidade, então algum dos produtos gerados é uma observação de avaliação.**

$$(\forall a_1) \text{ avaliacaoQualidade}(a_1) \rightarrow (\exists art_1) ( \text{observacaoAvaliacao}(art_1) \wedge \text{produto}(art_1, a_1) )$$

- **A6 – Se uma atividade de análise de medição  $a_1$  analisa um valor medido  $v_1$  produzido por atividade de medição  $med_1$ , então  $a_1$  depende de  $med_1$ .**

$$(\forall a_1, med_1, v_1) ( \text{analisa}(a_1, v_1) \wedge \text{produz}(med_1, v_1) ) \rightarrow \text{dependeDe}(a_1, med_1)$$

- **A7 – Se uma atividade é uma atividade de avaliação da qualidade, então ela depende de alguma atividade de análise de medição.**

$$(\forall a_1) \text{ avaliacaoQualidade}(a_1) \rightarrow (\exists a_2) ( \text{analiseMedicao}(a_2) \wedge \text{dependeDe}(a_1, a_2) )$$

#### 4.4.3.3 Avaliação da Sub-Ontologia de Avaliação

Assim como nas sub-ontologias de modelo de qualidade e de medição, a última etapa na definição da sub-ontologia de avaliação é a avaliação das suas questões de competência. Da mesma maneira que nas demais sub-ontologias, percebe-se que há questões cujas respostas são obtidas por uma única relação, como é o caso de QC1, que é respondida pela relação *é definido para* entre os conceitos *Modelo de Análise de Medição* e *Medida*. Porém, novamente há questões mais complexas que exigem vários

relacionamentos e axiomas para serem respondidas, como é o caso de QC9, para a qual são necessárias seis relações e quatro axiomas para respondê-la. As relações para responder QC9 são: (i) *é sub-tipo de* entre *Análise de Medição* e *Atividade*; (ii) *adota* entre *Atividade* e *Procedimento*; (iii) *é sub-tipo de* entre *Modelo de Análise de Medição* e *Procedimento*; (iv) *atribui* entre *Análise de Medição* e *Conclusão*; (v) *compõe* entre *Conclusão* e *Critério de Decisão*; e, por fim, (vi) *é definido por* entre *Modelo de Análise de Medição* e *Critério de Decisão*. Além das relações apresentadas, também são necessários os axiomas (A1), (A2), (A3) e (A4).

A Tabela 4.9 mostra a avaliação completa da sub-ontologia de avaliação.

**Tabela 4.9 – Avaliação da Sub-Ontologia de Avaliação.**

<b>Questão de Competência</b>	<b>Conceito A</b>	<b>Relação</b>	<b>Conceito B</b>	<b>Axiomas</b>
QC1	Modelo de Análise de Medição	é definido para	Medida	-
QC2	Modelo de Análise de Medição	é definido por	Critério de Decisão	-
QC3	Premissa	premissa de critério	Critério de Decisão	-
QC4	Conclusão	conclusão de critério	Critério de Decisão	-
QC5	Atividade	adota	Procedimento	A1, A2 e A3
	Análise de Medição	é sub tipo de	Atividade	
	Modelo de Análise de Medição	é sub tipo de	Procedimento	
	Análise de Medição	analisa	Valor Medido	
	Medição	produz	Valor Medido	
	Medição	aplica	Medida	
	Modelo de Análise de Medição	é definido para	Medida	

**Tabela 4.9 – Avaliação da Sub-Ontologia de Avaliação (continuação).**

<b>Questão de Competência</b>	<b>Conceito A</b>	<b>Relação</b>	<b>Conceito B</b>	<b>Axiomas</b>
QC6	Análise de Medição	analisa	Valor Medido	-
QC7	Atividade	depende de	Atividade	A6
	Análise de Medição	é sub tipo de	Atividade	
	Medição	é sub tipo de	Atividade	
	Análise de Medição	analisa	Valor Medido	
	Medição	produz	Valor Medido	
QC8	Análise de Medição	analisa	Valor Medido	A1
	Medição	produz	Valor Medido	
	Medição	aplica	Medida	
QC9	Análise de Medição	é sub-tipo de	Atividade	A1, A2, A3 e A4
	Atividade	adota	Procedimento	
	Modelo de Análise de Medição	é sub-tipo de	Procedimento	
	Análise de Medição	atribui	Conclusão	
	Conclusão	compõe	Critério de Decisão	
	Modelo de Análise de Medição	é definido por	Critério de Decisão	
QC10	Atividade	depende de	Atividade	A7
	Avaliação da Qualidade	é sub tipo de	Atividade	
	Análise de Medição	é sub tipo de	Atividade	

**Tabela 4.9 – Avaliação da Sub-Ontologia de Avaliação (continuação).**

Questão de Competência	Conceito A	Relação	Conceito B	Axiomas
CQ11	Artefato	produto	Atividade	A5
	Avaliação da Qualidade	é sub tipo de	Atividade	
	Observação de Avaliação	é sub tipo de	Artefato	
CQ12	Observação de Avaliação	é super tipo de	Ponto Fraco, Ponto Forte, Não Conformidade e Oportunidade de Melhoria	-
CQ13	Observação de Avaliação	gera	Ação de Qualidade	-
CQ14	Ação de Qualidade	é super tipo de	Ação Corretiva, Ação Preventiva e Ação de Melhoria	-

#### 4.4.3.4 Exemplos de Instanciação da Sub-Ontologia de Avaliação

A instanciação da sub-ontologia de avaliação é apresentada na Tabela 4.10 seguindo a mesma estrutura definida para as demais sub-ontologias. Um ponto importante a ser destacado é a necessidade de se mostrar dois exemplos dos conceitos de *medição* e de *valor medido* pertencentes à sub-ontologia de medição. Essa repetição torna-se necessária, uma vez que essas instâncias são a base para ilustrar a instanciação do conceito *modelo de análise de medição* pertencente à sub-ontologia de avaliação.



**Tabela 4.10 – Instanciação da Sub-Ontologia de Avaliação.**

<b>Medição 1</b>	Caracterizar o grau de implementação do resultado esperado de processo MPS.BR GPR 3 no Projeto P1	Avaliar a correção do uso do método da Especificação de Requisitos do Projeto P1
<b>Valor Medido 1</b>	L	0,7
<b>Medição 2</b>	Caracterizar o grau de implementação do resultado esperado de processo MPS.BR GPR 3 no Projeto P2	-
<b>Valor Medido 2</b>	T	-
<b>Modelo de Análise de Medição</b>	Modelo de Agregação dos resultados esperados dos processos e dos resultados esperados de atributos do processo nos projetos para se chegar à caracterização da unidade organizacional	Modelo de Análise do Grau de Adequação do Uso do Método na Elaboração da Especificação de Requisitos
<b>Critérios de Decisão</b>	<ul style="list-style-type: none"> <li>• CD1: &lt;premissa&gt; Todos X (isto é, todos T, ou todos L, ou todos P, ou todos N) → &lt;conclusão&gt; X</li> <li>• CD2: &lt;premissa&gt; Todos os projetos terminados X (isto é, todos T, ou todos L, ou todos P ou todos N) e os incompletos NA (Não Avaliado) → &lt;conclusão&gt; X</li> <li>• CD3: &lt;premissa&gt; Todos T ou L → &lt;conclusão&gt; L</li> <li>• CD4: &lt;premissa&gt; Todos T ou L e os incompletos NA (Não Avaliado) → &lt;conclusão&gt; L</li> <li>• CD5: &lt;premissa&gt; Existem P, mas não existem N (Pode existir NA - Não Avaliado) → &lt;conclusão&gt; L ou P</li> <li>• CD6: &lt;premissa&gt; Existe N → &lt;conclusão&gt; N, P ou L</li> <li>• CD7: &lt;premissa&gt; Resultado Esperado F (Fora do Escopo) → &lt;conclusão&gt; F</li> </ul>	<ul style="list-style-type: none"> <li>• CD1: &lt;premissa&gt; Valor Medido &lt; 0,4 → &lt;conclusão&gt; O método está sendo usado muito incorretamente e é necessário avaliar cuidadosamente as razões para tal.</li> <li>• CD2: &lt;premissa&gt; <math>0,4 \leq \text{Valor Medido} &lt; 0,8</math> → &lt;conclusão&gt; O método não está sendo usado de forma correta e é necessário avaliar em que pontos há falhas em sua aplicação.</li> <li>• CD3: &lt;premissa&gt; <math>0,8 \leq \text{Valor Medido} &lt; 1</math> → &lt;conclusão&gt; O método está sendo usado de maneira satisfatória, mas seu uso pode ser aperfeiçoado e, portanto, vale a pena avaliar em que pontos ainda há falhas em sua aplicação.</li> <li>• CD4: &lt;premissa&gt; Valor Medido = 1 → &lt;conclusão&gt; O método está sendo corretamente utilizado.</li> </ul>
<b>Premissa</b>	Ver em Critério de Decisão	Ver em Critério de Decisão
<b>Conclusão</b>	Ver em Critério de Decisão	Ver em Critério de Decisão

**Tabela 4.10 – Instanciação da Sub-Ontologia de Avaliação (continuação).**

<b>Análise de Medição</b>	Atribuir nível de implementação do REP MPS.BR GPR 3 na UO baseando-se nos valores medidos na medição de cada um dos projetos	Analisar a correção no uso do método para elaboração de especificação de requisitos
<b>Conclusão Atribuída</b>	L	O método não está sendo usado de forma correta e é necessário avaliar em que pontos há falhas em sua aplicação.
<b>Avaliação de Qualidade</b>	Avaliar a aderência ao modelo MPS.BR do REP MPS.BR GPR 3 com base na conclusão da análise de medição	Avaliar problemas na elaboração da especificação de requisitos com base na conclusão da análise de medição
<b>Observação de Avaliação</b>	O modelo de ciclo de vida de um projeto avaliado não estava aderente ao praticado no decorrer da realização do projeto	A notação do método não foi usada de maneira correta na elaboração da especificação de requisitos do Projeto 1
<b>Ponto Fraco</b>	O modelo de ciclo de vida de um projeto avaliado não estava aderente ao praticado no decorrer da realização do projeto	-
<b>Não Conformidade</b>	-	A notação do método não foi usada de maneira correta na elaboração da especificação de requisitos do Projeto 1
<b>Ação de Qualidade</b>	Criar um novo item no checklist de verificação da execução do projeto para avaliar se o ciclo de vida definido para o projeto está sendo seguido	Adequar a especificação de requisitos do Projeto 1 à notação proposta pelo método.
<b>Ação Preventiva</b>	Criar um novo item no checklist de verificação da execução do projeto para avaliar se o ciclo de vida definido para o projeto está sendo seguido	-
<b>Ação Corretiva</b>	-	Adequar a especificação de requisitos do Projeto 1 à notação proposta pelo método.

#### **4.5 Trabalhos Correlatos na Literatura**

Atualmente, muita atenção tem sido dada ao uso de ontologias na Engenharia de Software, com propósitos bastante variados. Em (CALERO et al., 2006), por exemplo, há o relato de diversos trabalhos dessa natureza, incluindo o desenvolvimento de ontologias para os domínios de manutenção e de medição de software, e a indicação da área de ambientes de desenvolvimento de software como um importante campo de

aplicação de ontologias na Engenharia de Software, o que corrobora o principal uso planejado para a Ontologia de Qualidade de Software proposta neste trabalho.

No que se refere a outros trabalhos apresentando ontologias no domínio de qualidade de software com foco em AMP, incluindo medição, três merecem destaque: (MARTIN e OLSINA, 2003), (BERTOA et al., 2006) e (DUARTE e FALBO, 2000).

MARTIN e OLSINA (2003) desenvolveram uma ontologia para métricas e indicadores de software para Web. Seu trabalho teve por objetivo formalizar e homogeneizar a conceituação nesse domínio, uma vez que há na literatura uma gama enorme de definições de métricas e indicadores para o domínio de medição de software para web. Para tal, sua ontologia está centrada em termos das normas internacionais ISO/IEC 15939 (ISO/IEC, 2003), ISO/IEC 9126-1 (ISO/IEC, 2001) e ISO/IEC 14598-1 (ISO/IEC, 1999).

BERTO A et al. (2006) desenvolveram uma ontologia para o domínio de medição de software. Segundo eles, a medição é uma disciplina relativamente jovem e, sendo assim, não existe um consenso sobre as definições de termos envolvidos nesse domínio, apontando, inclusive, inconsistência em normas e padrões. Seu trabalho buscou definir, então, uma terminologia coerente, consensual e amplamente aceita, a partir de uma exaustiva análise de diversos trabalhos, padrões e normas. Como resultado dessa análise, foram definidas quatro sub-ontologias: Caracterização e Objetivos da Medição de Software, que trata do estabelecimento do contexto e das metas da medição; Medidas de Software, que busca clarificar a terminologia envolvida nas definições de medidas; Abordagens de Medição, que descreve diferentes modos de obter resultados de medição; e Medição, que inclui conceitos relacionados à realização do processo de medição.

Conforme já citado no início da seção 4.4, DUARTE e FALBO (2000) desenvolveram uma ontologia de qualidade de software sub-dividida em duas sub-ontologias: Características de Qualidade – Natureza e Estrutura, fortemente baseada na ISO 9126 e que trata dos tipos de características de qualidade e sua aplicação; e Qualidade – Como Medir, que de modo simplificado trata da medição de software.

Todos esses trabalhos, de alguma forma, foram inspiração para a Ontologia de Qualidade de Software aqui desenvolvida. Entretanto, suas conceituações não foram integralmente reutilizadas, principalmente porque parecem muito voltadas para avaliação de produtos de software.

Dentre os trabalhos citados, o que mais influenciou a ontologia aqui desenvolvida foi o de BERTOIA et al. (2006). A sub-ontologia de Modelo de Qualidade corresponde à sub-ontologia de Caracterização e Objetivos da Medição de Software de BERTOIA et al. A percepção de que há entidades e tipos de entidades está presente em ambas as ontologias. Além disso, ambas tratam de elementos que caracterizam os tipos de entidades, mas na sub-ontologia de Modelo de Qualidade há uma distinção a mais, indicando que elementos podem ser direta ou indiretamente mensuráveis. Por outro lado, alguns conceitos, como Modelo de Qualidade, Necessidade de Informação e Conceito Mensurável, não foram incorporados. Em relação à sub-ontologia de Medição, que corresponde às demais sub-ontologias de (BERTOIA et al., 2006), há também algumas diferenças: (i) preferiu-se não considerar indicador como um tipo de medida como em (BERTOIA et al., 2006), pois se considerou que ser um indicador é um papel de uma medida e não um sub-tipo; (ii) uma vez que é feita uma distinção entre elementos direta e indiretamente mensuráveis, optou-se por relacionar medidas base e derivadas diretamente como esses dois conceitos, respectivamente; (iii) como a ontologia aqui desenvolvida foi construída de forma integrada com a ontologia de processo de software, a parte que trata de abordagens de medição em (BERTOIA et al., 2006) foi adaptada para ficar em linha com a ontologia de processo de ODE.

Das sub-ontologias apresentadas, percebe-se que a principal contribuição da Ontologia de Qualidade de Software desenvolvida está na sub-ontologia de avaliação, pois não foi encontrado nenhum trabalho correlato na literatura.

#### ***4.6 Considerações Finais do Capítulo***

Neste capítulo foi apresentada a Ontologia de Qualidade de Software com foco na Avaliação e Melhoria de Processos (AMP) desenvolvida neste trabalho. Foram apresentados, ainda, exemplos de instanciação da ontologia tanto para a perspectiva de processo quanto para a perspectiva de produto de software.

Vale ressaltar que ontologias são importantes para o apoio no desenvolvimento de ferramentas, tanto como uma especificação reutilizável quanto como um mecanismo para favorecer a interoperabilidade entre as ferramentas desenvolvidas. Assim, a ontologia apresentada neste capítulo é usada como base para o desenvolvimento de ferramentas de apoio à AMP, apresentadas no próximo capítulo desta dissertação.

### **Apoio Automatizado ao Processo de Avaliação e Melhoria de Processos de Software**

O objetivo deste capítulo é apresentar as ferramentas desenvolvidas neste trabalho para prover apoio automatizado ao processo de Avaliação e Melhoria de Processos de Software (AMP) no ambiente ODE (*Ontology-based Software Development Environment*), um ambiente de desenvolvimento de software centrado em processo e baseado em ontologias. Tais ferramentas foram desenvolvidas com base na Ontologia de Qualidade de Software e no processo de AMP definido.

#### **5.1 Introdução**

O processo de Avaliação e Melhoria de Processos de Software (AMP) possui algumas características marcantes, entre elas, destacam-se a sua complexidade, abrangência e uma forte inter-relação com os demais processos do ciclo de vida de software.

Analisando essas características, percebe-se que uma organização que busca a melhoria contínua de seus processos deverá, via de regra, despender considerável esforço para alcançar êxito nessa tarefa, que não é trivial de ser realizada. Logo, prover apoio automatizado ao processo de AMP pode ser considerado como um grande aliado e até mesmo um fator chave para as organizações implementarem com sucesso a melhoria contínua de seus processos.

Mais ainda, analisando, sobretudo, a questão da sua complexidade e inter-relação com os demais processos, percebe-se que prover apoio automatizado ao processo de AMP requer, geralmente, diversas ferramentas e, além disso, elas devem ser integradas entre si. Nesse contexto, uma solução possível é o uso de Ambientes de Desenvolvimento de Software (ADS), pois eles se colocam como uma possibilidade de integração de ferramentas para apoiar de maneira consistente, além do processo de AMP, os demais processos com os quais ele se relaciona, apoiando, assim, o processo de software como um todo, ou pelo menos porções significativas dele (HARRISON et al., 2000).

Porém, mesmo utilizando um ambiente com ferramentas integradas, é possível que elas sejam desenvolvidas por equipes diferentes que podem tomar por base diferentes conceituações. Uma forma de homogeneizar essas conceituações é a utilização de ontologias como base para especificação conceitual comum entre as ferramentas.

Pensando nisso, para apoiar, mesmo que parcialmente, o processo de AMP segundo as características acima, decidiu-se desenvolver ferramentas no contexto do Projeto ODE (*Ontology-based software Development Environment*) (FALBO et al., 2003), um ambiente de desenvolvimento de software centrado em processo e baseado em ontologias.

Este capítulo visa apresentar as ferramentas desenvolvidas e está estruturado da seguinte forma: a Seção 5.2 fornece uma visão geral do ambiente ODE, descrevendo suas principais características e ferramentas; a Seção 5.3 apresenta a ferramenta AvaliaODE, responsável pelo apoio à avaliação de processos e produtos de software; a Seção 5.4 apresenta a ferramenta MelhoriaODE, responsável pelo apoio à melhoria de processos de software; e, por fim, a Seção 5.5 apresenta as considerações finais do capítulo.

## ***5.2 O Ambiente de Desenvolvimento de Software ODE***

O ambiente ODE (*Ontology-based software Development Environment*), (FALBO et al., 2003) é um ADS centrado em processo, que tem sua fundamentação baseada em ontologias. Com base nessa fundamentação, as ferramentas de ODE são construídas, pois se acredita que a integração pode ser facilitada pelo fato dos conceitos envolvidos estarem bem definidos pelas ontologias.

Desse modo, a premissa do Projeto ODE está baseada no argumento de que, se as ferramentas em um ADS são construídas baseadas em ontologias, a integração delas pode ser obtida de forma mais fácil. A mesma ontologia pode ser usada para construir diferentes ferramentas, que apóiam atividades de engenharia de software correlacionadas. Assim, as ontologias nas quais ODE se baseia têm sido construídas procurando reutilizar conceituações previamente estabelecidas por outras ontologias já desenvolvidas no projeto, formando uma grande rede de conceitos relacionados à Engenharia de Software (BERTOLLO, 2006).

Atualmente a base ontológica de ODE conta com diversas ontologias, dentre elas: Ontologia de Processo de Software (BERTOLLO, 2006), Ontologia de Gerência

de Configuração de Software (ARANTES et al., 2007), Ontologia de Requisitos de Software (NARDI, 2006), Ontologia de Organizações de Software (RUY, 2006) e Ontologia de Riscos de Software (FALBO et al., 2004).

Construídas a partir dessa base ontológica, há ferramentas para apoiar a definição de processos de software (BERTOLLO et al., 2006), acompanhamento de projetos (MORO et al., 2005), gerência de riscos (FALBO et al., 2004a), gerência de configuração (NUNES et al., 2006), garantia da qualidade (PALACIO, 2007) e engenharia de requisitos (MARTINS et al., 2006), dentre outras.

Entretanto ODE não conta com ferramentas específicas para apoiar o processo de Avaliação e Melhoria de Processos de Software (AMP). Especificamente em relação à avaliação, a ferramenta de apoio ao processo de Garantia da Qualidade, GQA-ODE (PALACIO, 2007), possui funcionalidades que apóiam de maneira simplificada uma avaliação no contexto da garantia da qualidade. Contudo, ela não contempla diversos aspectos a serem tratados na avaliação de processos e produtos de software, o que motiva a construção de ferramentas tanto para avaliação quanto para melhoria de processos. Assim, foram desenvolvidas neste trabalho as ferramentas AvaliaODE, responsável pela avaliação de processos e produtos de software, e MelhoriaODE, responsável pela melhoria de processos em ODE, apresentadas na sequência.

### ***5.3 AvaliaODE: Uma Ferramenta de Apoio à Avaliação de Processos***

Apoiar a avaliação de processos e produtos de software em ODE, ainda que parcialmente, é o objetivo principal da ferramenta AvaliaODE. No Capítulo 3 desta dissertação foram identificadas funcionalidades para prover apoio automatizado ao processo de AMP. Contudo, apenas algumas delas foram selecionadas como parte do escopo da ferramenta AvaliaODE, a saber: (i) apoio à elaboração de uma proposta de escopo para a avaliação de um processo, (ii) apoio à avaliação de um processo segundo critérios definidos, (iii) registro de conclusões de uma avaliação de processo e (iv) apoio à avaliação de produtos, de forma análoga à avaliação de processos.

Uma vez estabelecido o escopo a ser trabalhado, foram definidos os casos de uso que descrevem de forma mais detalhada essas funcionalidades. A Tabela 5.1 apresenta a correlação entre as funcionalidades identificadas e os casos de uso que as implementam.

**Tabela 5.1 – Correlação entre os requisitos funcionais e os casos de uso de AvaliaODE.**

Funcionalidade	Caso de Uso
FE3 – Apoio à elaboração de uma proposta de escopo para a avaliação de um processo	Definir Escopo de Avaliação de Processo
FE4 – Apoio à avaliação de um processo segundo critérios definidos	Definir Critérios de Avaliação de Processo, Cadastrar <i>Checklist</i> , Avaliar Processo, Realizar Atividade de Avaliação da Qualidade, Responder <i>Checklis</i> e Controlar Não Conformidade
FE5 – Registro de conclusões de uma avaliação de processo	Avaliar Processo Controlar Não Conformidade
FE11 – Apoio à avaliação de produtos	Definir Critérios de Avaliação de Processo, Cadastrar <i>Checklist</i> , Avaliar Processo, Realizar Atividade de Avaliação da Qualidade, Responder <i>Checklis</i> e Controlar Não Conformidade

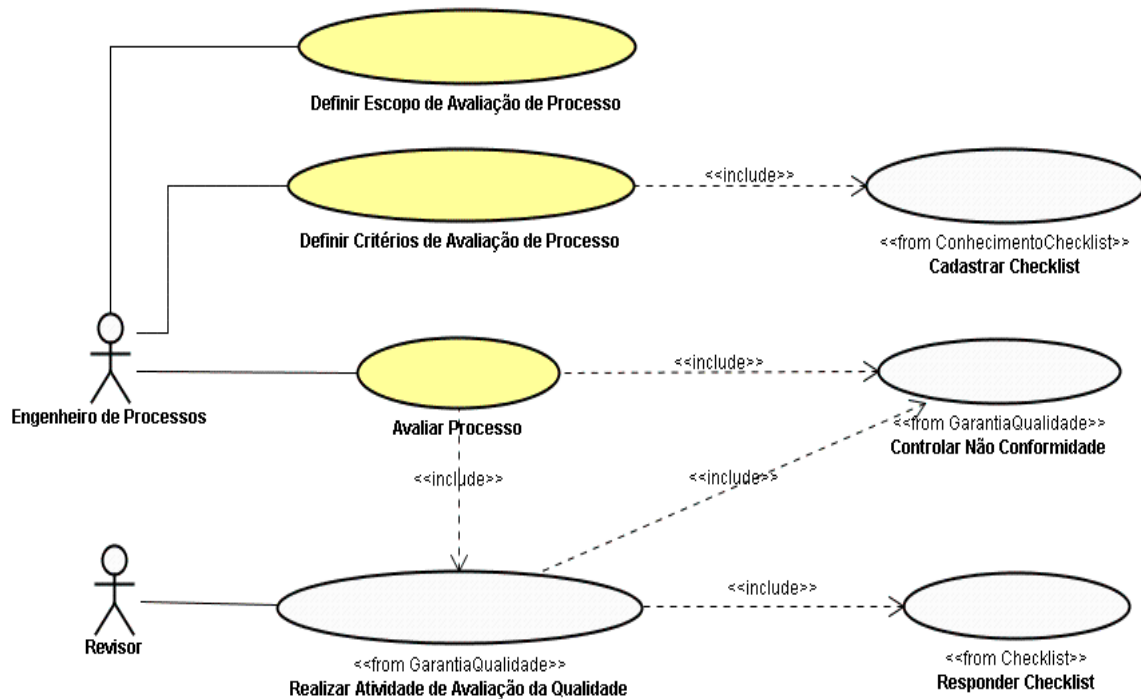
#### 5.4.1 – Modelo de Casos de Uso de AvaliaODE

A Figura 5.1 mostra o diagrama de casos de uso da ferramenta AvaliaODE. Nesse diagrama estão presentes dois atores: o Revisor, que representa o papel desempenhado pelos desenvolvedores responsáveis pelas avaliações da qualidade de produtos e processos, e o Engenheiro de Processos, que representa o papel desempenhado pelos responsáveis pela área de processos de uma organização.

Os casos de uso específicos da ferramenta AvaliaODE estão em destaque na Figura 5.1, a saber: *Definir Escopo de Avaliação de Processo*, *Definir Critérios de Avaliação de Processo* e *Avaliar Processo*. Os demais casos de uso são funcionalidades de GQA-ODE, a ferramenta de apoio ao processo de Garantia da Qualidade de ODE. O caso de uso *Cadastrar Checklist* é utilizado para a definição de critérios de avaliação de processos e produtos de software. O caso de uso *Realizar Atividade de Avaliação da Qualidade* é utilizado neste contexto para realizar os casos de uso *Responder Checklist*, responsável pelo preenchimento do *checklist* de um elemento avaliado, e *Controlar Não Conformidade*, que também pode ser realizado pelo caso de uso *Avaliar Processo*, tendo como finalidade o registro e tratamento de uma não conformidade identificada durante



uma avaliação de um processo ou produto de software. A seguir, os casos de uso específicos da ferramenta AvaliaODE são descritos sucintamente.



**Figura 5.1 – Diagrama de Casos de Uso de AvaliaODE.**

- **Definir Escopo de Avaliação de Processo:** Este caso de uso é responsável pela definição do escopo de uma avaliação de um processo organizacional. Seu objetivo é permitir que o engenheiro de processos escolha o processo padrão que será avaliado e, opcionalmente, seus artefatos. Ele também deve indicar quais dos projetos que instanciaram o processo padrão serão usados na avaliação. Além disso, para que a avaliação aconteça de fato, ele deve finalizar a definição do escopo da avaliação. Uma vez definido o escopo a ser avaliado, são, então, criadas uma avaliação de processo para cada projeto que terá o processo instanciado avaliado e uma avaliação de produto para cada artefato do tipo selecionado produzido em cada projeto.
- **Definir Critérios de Avaliação de Processo:** Este caso de uso é responsável pela definição de um *checklist* para cada elemento do escopo de uma avaliação (processo padrão ou artefato) que será objeto de análise em uma avaliação de processo. Seu objetivo é permitir que o engenheiro de processos selecione uma avaliação de processo com escopo definido para o qual deseja definir os critérios

de avaliação. Como critério de avaliação foi adotado o uso de *checklists*. Assim, o engenheiro de processos deve selecionar um dos elementos do escopo de avaliação (processo ou tipo de artefato) e indicar um critério de avaliação na forma de um *checklist*. Por fim, o engenheiro de processos pode selecionar um *checklist* existente ou elaborar um novo *checklist* realizando o caso de uso “Cadastrar *Checklist*”.

- **Avaliar Processo:** Este caso de uso é responsável pela realização de uma avaliação de processo. Vale destacar que a realização da avaliação de processo se dá por meio da realização das atividades de avaliação da qualidade dos elementos (processo padrão e artefatos) definidos no escopo da avaliação de processo. Desse modo, o objetivo desse caso de uso é permitir que o engenheiro de processos selecione a avaliação de processo e, em seguida, uma de suas atividades de avaliação de qualidade para ser realizada. A atividade pode, então, ser realizada, por meio do caso de uso “Realizar Atividade de Avaliação da Qualidade”. Uma vez realizada uma atividade de avaliação da qualidade, o engenheiro de processos pode registrar observações para ela. Além disso, uma vez que tenham sido realizadas todas as atividades de avaliação da qualidade e, também, existam observações para todas elas, o engenheiro de processos pode, então, relatar um parecer final sobre a avaliação de processo.

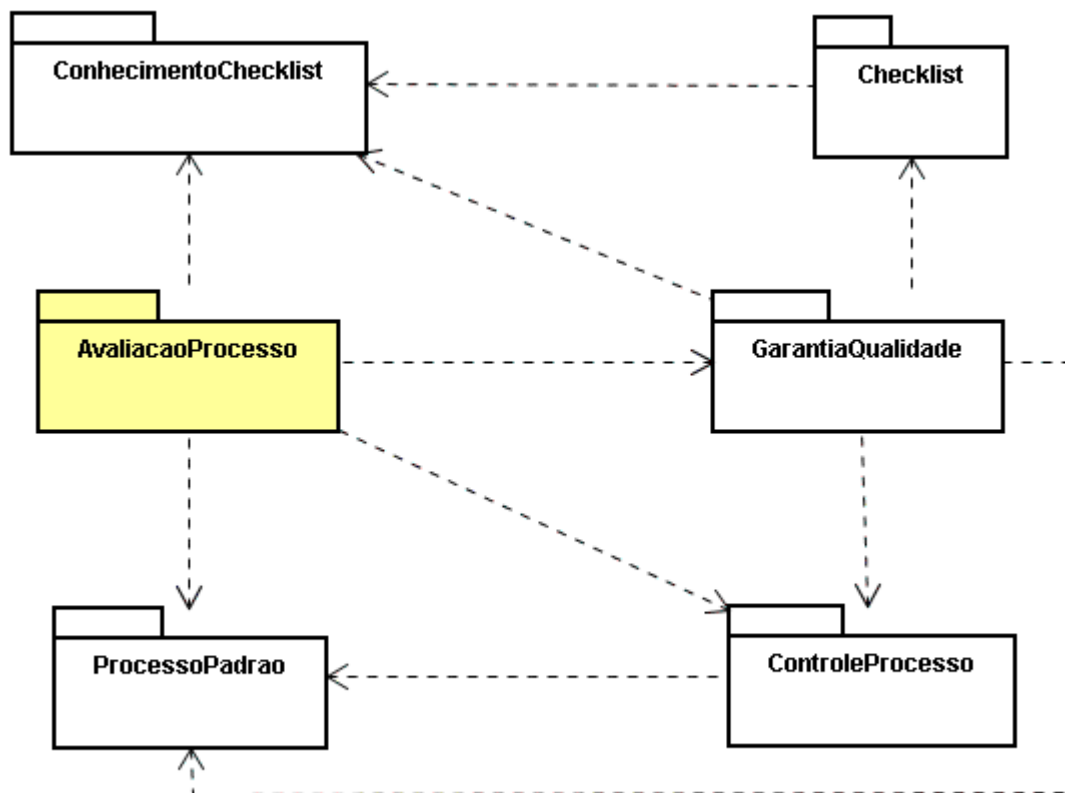
#### 5.4.2 – Modelo Conceitual de AvaliaODE

Uma vez definidos os casos de uso da ferramenta, deve-se elaborar o modelo conceitual para tratá-los. A Figura 5.2 apresenta o diagrama de pacotes envolvendo os pacotes que tratam das funcionalidades relacionadas ao apoio à avaliação de processos.

O pacote *AvaliacaoProcesso* é o pacote principal da ferramenta. Ele contém as classes referentes à avaliação de processos e produtos de software e depende dos demais pacotes mostrados na figura para permitir a definição do escopo de uma avaliação de processo (pacotes *ConhecimentoChecklist*, *ProcessoPadrao* e *ControleProcesso*), a definição de *checklists* (pacote *ConhecimentoChecklist*) e para a realização de atividades de avaliação da qualidade (pacotes *GarantiaQualidade* e *ControleProcesso*), o que pode envolver responder *checklists* (pacote *Checklist*).

O pacote *ConhecimentoChecklist* contém as classes relacionadas com a definição de *checklists*, incluindo seus critérios de avaliação. No pacote *Checklist* estão

as classes relacionadas com o preenchimento de *checklists* em atividades de avaliação específicas. Já o pacote *ProcessoPadrao* contém as classes relacionadas com a definição dos processos padrão de uma organização. Esses processos servem de base para a definição de processos para projetos específicos, que consideram as características particulares do projeto. O pacote *ControleProcesso* contém as classes que modelam a definição de processos de projeto. O pacote *GarantiaQualidade*, por sua vez, contém as classes referentes às atividades de avaliação da qualidade e ao controle de não conformidades de ODE.



**Figura 5.2 – Diagrama de Pacotes de AvaliaODE.**

As figuras 5.3 e 5.4 apresentam diagramas de classes parciais do pacote *AvaliacaoProcesso*. As classes em destaque fazem parte do pacote *AvaliacaoProcesso* e foram inseridas como contribuição deste trabalho. Uma vez que esse pacote depende de outros pacotes, é necessário descrever algumas das classes dos mesmos que são utilizadas pelo primeiro, sobretudo aquelas do pacote *ProcessoPadrao* e *ControleProcesso*.

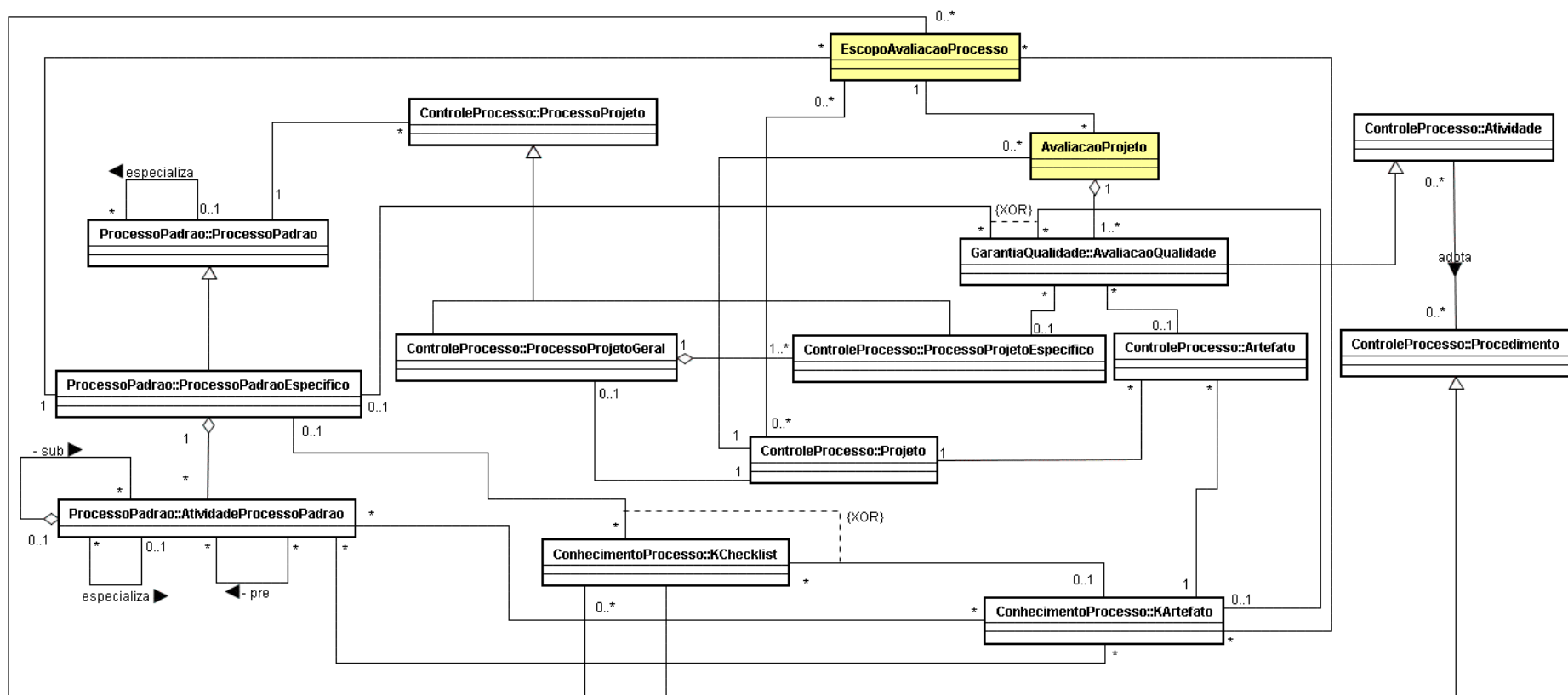


Figura 5.3 – Diagrama de Classes de AvaliaODE: Definição do Escopo de uma Avaliação.

Em ODE, há dois tipos de processos padrão (classe *ProcessoPadrao*): gerais (*ProcessoPadraoGeral*), representando processos padrão completos de uma organização, tal como um processo padrão para o desenvolvimento de aplicações Web, compostos de diversos sub-processos, definidos para uma finalidade específica, ditos processos específicos (*ProcessoPadraoEspecifico*), que representam sub-processos padrão específicos, tal como um processo padrão de Gerência de Projetos. Processos padrão específicos são decompostos em atividades (*AtividadeProcessoPadrao*). Especificamente na ferramenta AvaliaODE apenas processos padrão específicos podem ser selecionados para serem avaliados. Contudo, na ferramenta MelhorioODE, apresentada na próxima seção, ambas as sub-classes de *ProcessoPadrao* são utilizadas.

Um processo de projeto é definido para um projeto (classe *Projeto*), tomando por base um processo padrão (classe *ProcessoPadrao*). De forma análoga aos processos padrão, há a instanciação de processo padrão gerais, os ditos processos de projeto gerais (classe *ProcessoProjetoGeral*), que são decompostos em processos de projeto específicos (classe *ProcessoProjetoEspecifico*), estes decompostos em atividades (classe *Atividade*) e que instanciam processos padrão específicos.

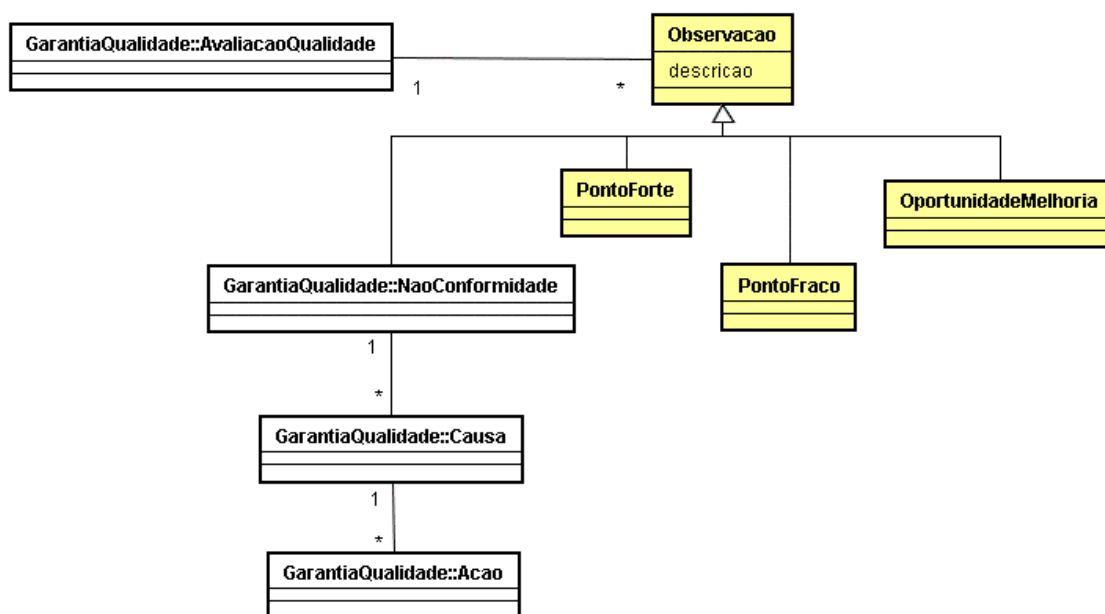
O ponto de partida para a avaliação de um processo é a definição do escopo dessa avaliação (classe *EscopoAvaliacaoProcesso*). No momento da definição do escopo, devem ser identificados o processo padrão a ser avaliado (classe *ProcessoPadraoEspecifico*), os projetos (classe *Projeto*) que o instanciaram e cujos processos serão avaliados (classe *ProcessoProjeto*) e, se necessário, os tipos de artefatos (classe *KArtefato*) que também devem ser considerados no escopo da avaliação do processo padrão.

Uma vez definido o escopo, são geradas avaliações de projeto (classe *AvaliacaoProjeto*) para avaliar o processo padrão em cada um dos projetos selecionados bem como para avaliar cada um dos artefatos, pertinentes ao escopo definido, produzidos pelos mesmos. A seguir, para cada um dos elementos do escopo (processo padrão e artefatos) é criada uma atividade de avaliação da qualidade (classe *AvaliacaoQualidade*).

Outra funcionalidade de AvaliaODE é a definição de critérios de avaliação na forma de *checklists*. Para cada atividade de avaliação da qualidade, deve ser definido um *checklist* (classe *KChecklist*) que será respondido no momento da realização da mesma. Este último aspecto (a resposta a *checklists*) não está retratado no modelo da Figura 5.3, podendo ser encontrado em (PALÁCIO, 2007).

Como mostra a Figura 5.4, após uma atividade de avaliação da qualidade ser realizada, o engenheiro de processos pode registrar observações acerca dessa avaliação (classe *Observação*), que podem ser pontos fortes (classe *PontoForte*), pontos fracos (classe *PontoFracos*), oportunidades de melhoria (classe *OportunidadeMelhoria*) ou não conformidades (classe *NaoConformidade*). Caso seja uma não conformidade, o engenheiro de processos pode, ainda, identificar causas (classe *Causa*) e propor ações (classe *Acao*) para solucioná-las.

Vale ressaltar que os diagramas de classes apresentados nas figuras 5.3 e 5.4 são parciais. O modelo completo da ferramenta AvaliaODE pode ser encontrado em (BRINGUENTE, 2008).

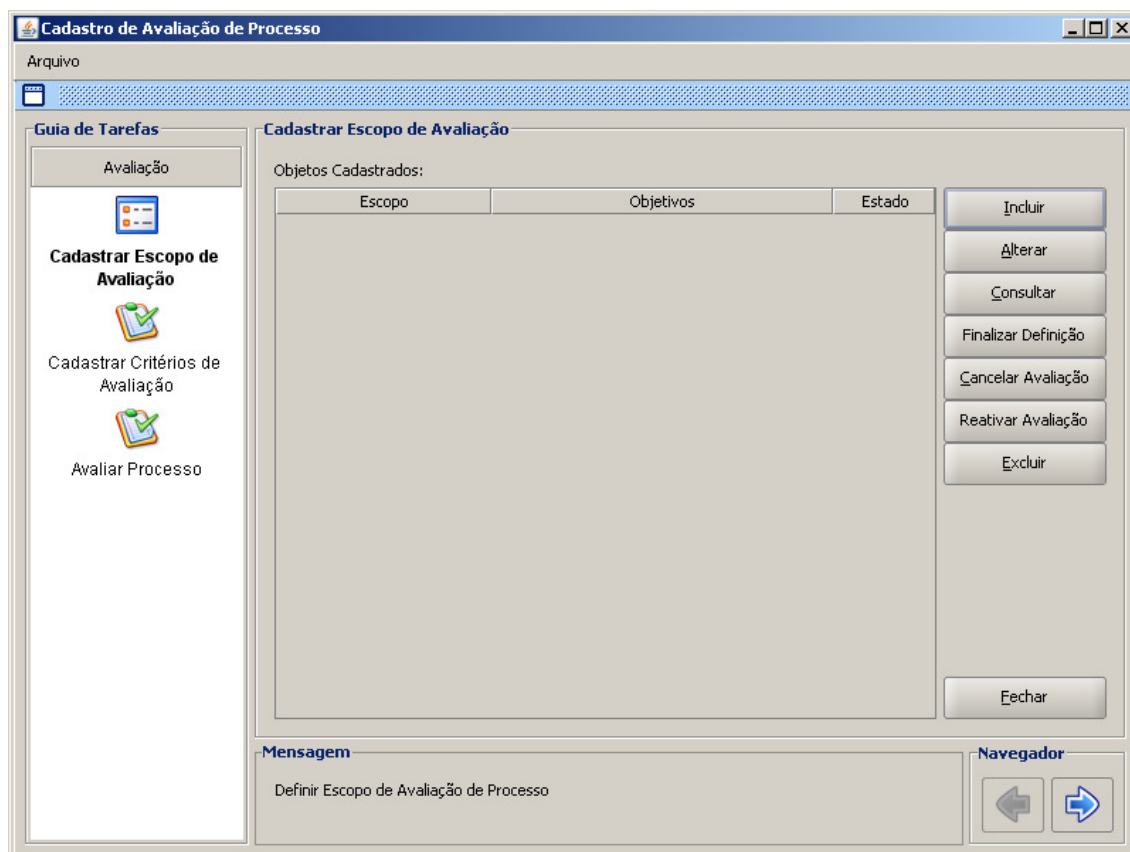


**Figura 5.4 – Diagrama de Classes de AvaliaODE: Resultados da Avaliação.**

### 5.3.3 – Apresentação de AvaliaODE

Nesta seção são apresentadas as principais funcionalidades da ferramenta AvaliaODE, por meio da apresentação de algumas de suas interfaces.

A Figura 5.5 apresenta a tela principal da ferramenta. Seguindo os padrões do ambiente ODE, essa interface conta com um menu no canto esquerdo contendo as principais atividades apoiadas pela ferramenta. Ao selecionar um item do menu, um painel referente ao item selecionado é mostrado à direita. Na parte inferior da tela existe um navegador que possibilita avançar ou retroceder nessas atividades.



**Figura 5.5 – Tela principal de AvaliaODE.**

Uma vez selecionado o item de menu *Definir Escopo de Avaliação*, é permitido ao engenheiro de processos incluir, alterar, consultar ou excluir um escopo de avaliação, bem como finalizar sua definição. Uma vez definido o escopo, o engenheiro de processos pode cancelá-lo, impossibilitando que sejam criados critérios de avaliação para ele, ou reativá-lo, caso algum escopo tenha sido cancelado por engano.

Para ilustrar a utilização da ferramenta, as figuras 5.6, 5.7 e 5.8 mostram a inclusão de um escopo de avaliação. Conforme pode ser observado na Figura 5.6, o engenheiro de processos deve indicar primeiramente o processo padrão geral e o subprocesso padrão específico que ele deseja avaliar. Além disso, ele deve informar uma descrição sucinta do escopo e também informar quais são os objetivos dessa avaliação. A seguir, conforme mostrado na Figura 5.7, o engenheiro de processos deve indicar quais tipos de artefatos do processo padrão específico selecionado devem fazer parte do escopo de avaliação. Por fim, conforme mostrado na Figura 5.8, ele deve selecionar quais projetos, dentre aqueles que instanciaram o processo padrão específico a ser avaliado, devem fazer parte do escopo dessa avaliação.

**Escopo de Avaliação de Processo**

Dados Gerais   Artefatos   Projetos

Processo Padrão: Processo Padrão LabES ▼

Sub-Processo: Desenvolvimento de Software ▼

Descrição sucinta:

Avaliação de Atividades de Engenharia de Requisitos do Processo de Desenvolvimento

Objetivo:

O objetivo dessa avaliação é verificar se o processo de desenvolvimento de software do Processo Padrão LabES está sendo adequadamente seguido nos projetos da organização e apontar oportunidades de melhoria, sobretudo, nos documentos de especificação de requisitos e de análise.

Ok   Fechar

**Figura 5.6 – Definição de Escopo de Avaliação em AvaliaODE – Dados Gerais.**

**Escopo de Avaliação de Processo**

Dados Gerais   Artefatos   Projetos

Selecione os artefatos produzidos pelo sub-processo

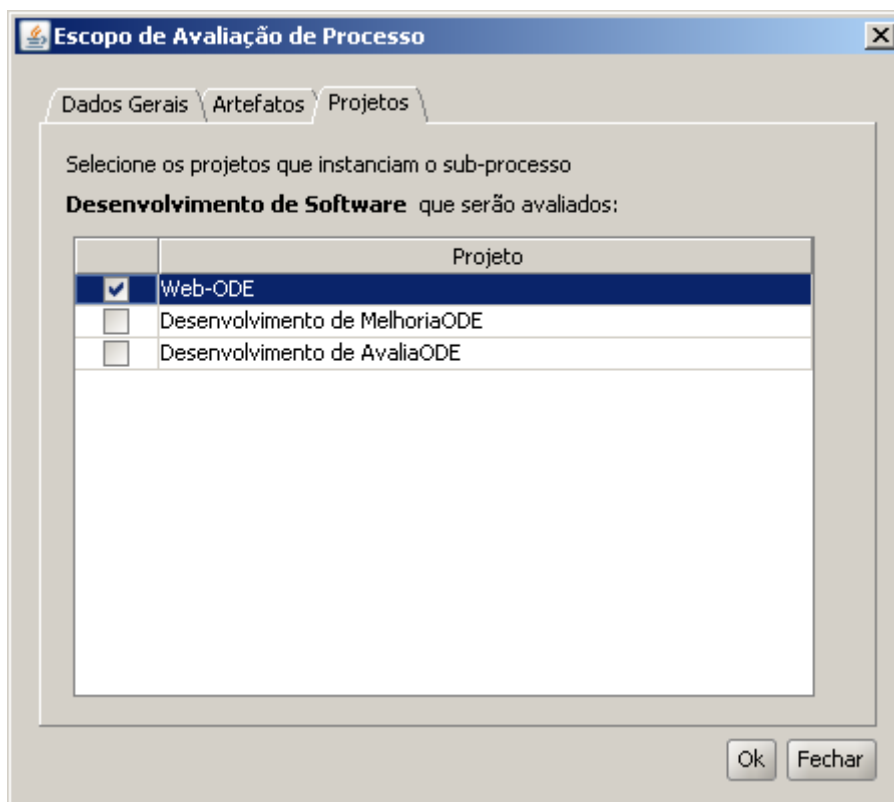
**Desenvolvimento de Software** que serão avaliados:

	Artefato	Atividade Produtora
<input checked="" type="checkbox"/>	Documento de Especificação de Requisitos	Análise de Requisitos
<input checked="" type="checkbox"/>	Documento de Especificação de Análise	Análise de Requisitos
<input type="checkbox"/>	Documento de Testes	Testes
<input type="checkbox"/>	Documento de Especificação de Projeto	Projeto
<input type="checkbox"/>	Documento de Implementação	Implementação
<input type="checkbox"/>	Produto Implementado	Implementação

Ok   Fechar

**Figura 5.7 – Definição de Escopo de Avaliação em AvaliaODE – Seleção de Tipos de Artefatos.**





**Figura 5.8 – Definição de Escopo de Avaliação em AvaliaODE – Seleção de Projetos.**

Com base em um escopo de avaliação definido, o engenheiro de processos pode, então, selecionar um *checklist*, como critério de avaliação, para cada um dos elementos do escopo de avaliação (processo padrão e artefatos) dos projetos selecionados na definição do escopo de avaliação.

Conforme pode ser observado na Figura 5.9, o engenheiro de processos deve inicialmente indicar para qual avaliação de processo ele quer cadastrar os *checklists*. Uma vez selecionada a avaliação, o engenheiro de processos deve indicar o tipo do elemento, artefato ou sub-processo, para o qual ele irá definir o *checklist*. Após escolher o tipo, ele deve selecionar o próprio elemento e, em seguida, ele pode elaborar um novo *checklist* ou selecionar um *checklist* já cadastrado para o tipo selecionado. Por fim, ao cadastrar ou selecionar o *checklist* ele é adicionado na lista de *checklists* da avaliação de processo selecionada. Pode ser observado na figura 5.9 que há dois *checklists*, sendo o primeiro referente ao produto Documento de Especificação de Análise e o segundo referente ao sub-processo Desenvolvimento de Software, ambos pertencentes ao projeto Web-ODE.

Selecione uma avaliação de processo: Avaliação do Documento de Especificação de Análise ▼

Selecione o tipo de elemento: ☒ Artefato ☐ Sub-Processo

Selecione o elemento: Documento de Especificação de Análise ▼

Critérios da avaliação de processo :

Checklist	Item
Cheklist de Avaliação de Documento de Esp...	Documento de Especificaç...
Checklist de Avaliação do Processo de Dese...	Desenvolvimento de Soft...

Elaborar Checklist

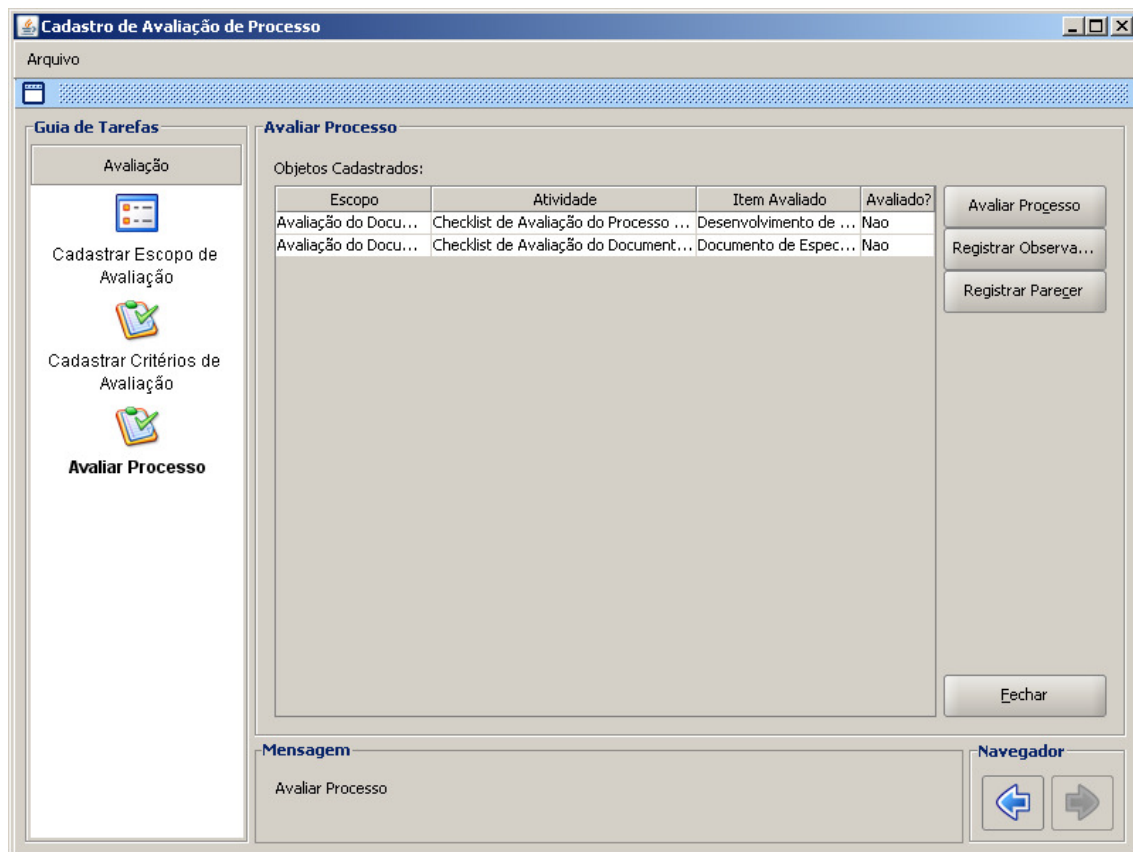
Selecionar Checklist

Cancelar

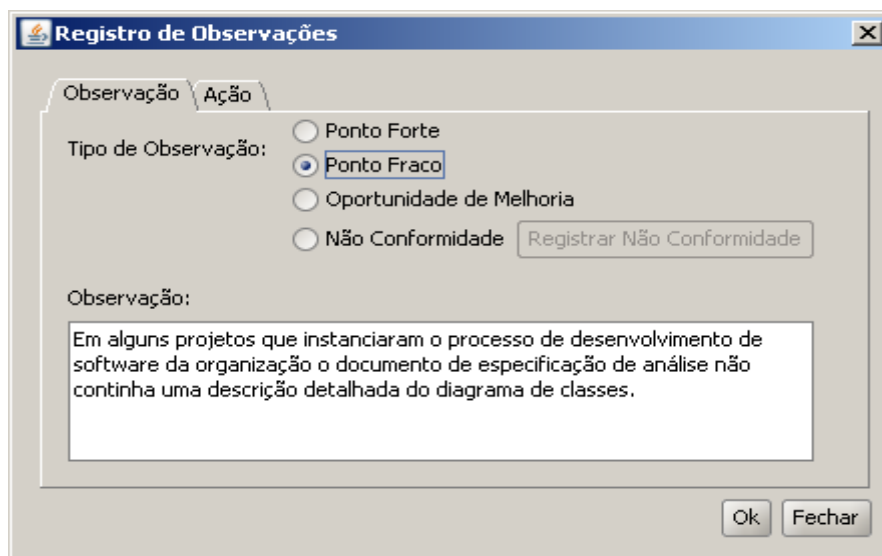
Ok Fechar

**Figura 5.9 – Definição de Critérios de Avaliação em AvaliaODE.**

Uma vez definidos os *checklists* como critérios de avaliação, o engenheiro de processos pode avaliar os itens do escopo de avaliação. A Figura 5.10 ilustra duas atividades de avaliação da qualidade que ainda não foram realizadas. Para realizá-las, basta que sejam respondidos os correspondentes *checklists* definidos como critério de avaliação para os respectivos itens. No exemplo mostrado na Figura 5.10 há um *checklist* para a avaliação do Processo de Desenvolvimento de Software e outro para o Documento de Especificação de Análise. Após realizar uma atividade de avaliação da qualidade, ou seja, preencher o *checklist* de um item do escopo de avaliação, o engenheiro de processos pode, então, registrar observações (Figura 5.11).

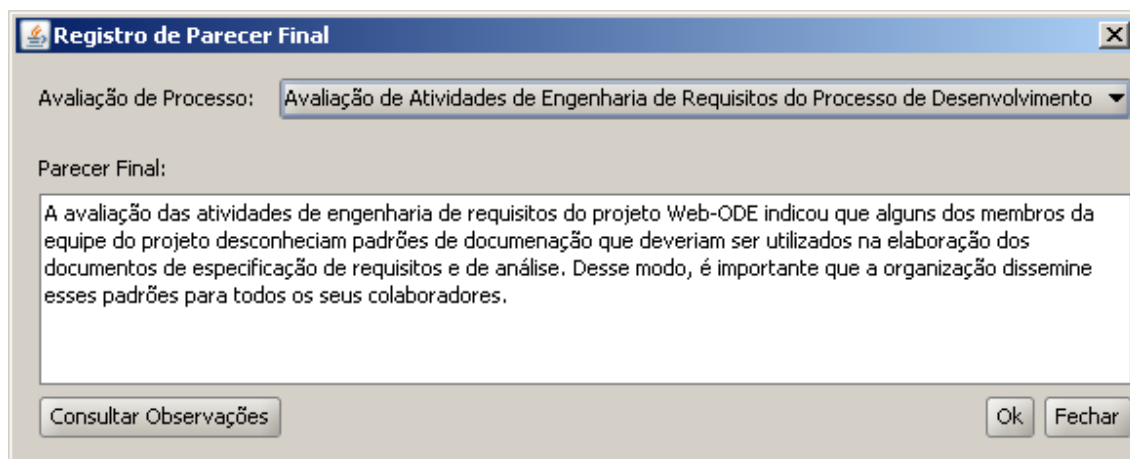


**Figura 5.10 – Definição de Critérios de Avaliação em AvaliaODE.**



**Figura 5.11 – Registro de Observação de Avaliação em AvaliaODE.**

Por fim, após realizar todas as atividades de avaliação da qualidade de uma avaliação de processo e registrar pelo menos uma observação para cada uma das atividades realizadas, o engenheiro de processos pode dar um parecer final sobre a avaliação de processo realizada, conforme é ilustrado pela Figura 5.12, encerrando, assim, a avaliação e não sendo mais permitidas alterações na mesma.



**Figura 5.12 – Registro de Parecer Final de Avaliação em AvaliaODE.**

#### ***5.4 MelhoriaODE - Uma Ferramenta de Apoio à Melhoria de Processos***

A ferramenta MelhoriaODE tem por objetivo principal apoiar, ainda que parcialmente, a melhoria de processos em ODE. O escopo definido para ser tratado no contexto deste trabalho contempla apenas algumas das funcionalidades identificadas no Capítulo 3, a saber: (i) apoio à coleta de informações que servirão de base para a melhoria de um processo, (ii) apoio à definição de uma proposta de melhoria do processo e (iii) apoio ao registro de resultados e lições aprendidas com a melhoria de processos.

Uma vez estabelecido o escopo a ser trabalhado, foram definidos os casos de uso que descrevem de forma mais detalhada essas funcionalidades. A Tabela 5.2 apresenta a correlação entre as funcionalidades identificadas e os casos de uso que as implementam.

**Tabela 5.2 – Correlação entre os requisitos funcionais e os casos de uso de MelhoriaODE.**

Funcionalidade	Caso de Uso
FE1 – Apoio à coleta de informações que servirão de base para a melhoria de um processo	Cadastrar Sugestão de Melhoria de Processos
FE7 – Apoio à definição da proposta de melhoria do processo	Elaborar Proposta de Melhoria de Processos e Avaliar Processo
FE13 – Apoio ao registro de resultados e lições aprendidas com a melhoria de processos	Registrar Resultados de Melhoria de Processo e Cadastrar Lição Aprendida

### 5.4.1 – Modelo de Casos de Uso de MelhoriaODE

A Figura 5.13 mostra o diagrama de casos de uso da ferramenta MelhoriaODE. Nesse diagrama estão presentes dois atores: o Desenvolvedor, que representa todos os profissionais atuando nas atividades de desenvolvimento, tais como analistas, projetistas e programadores, e o Engenheiro de Processos, que representa o papel desempenhado pelos responsáveis pela área de processos de uma organização.

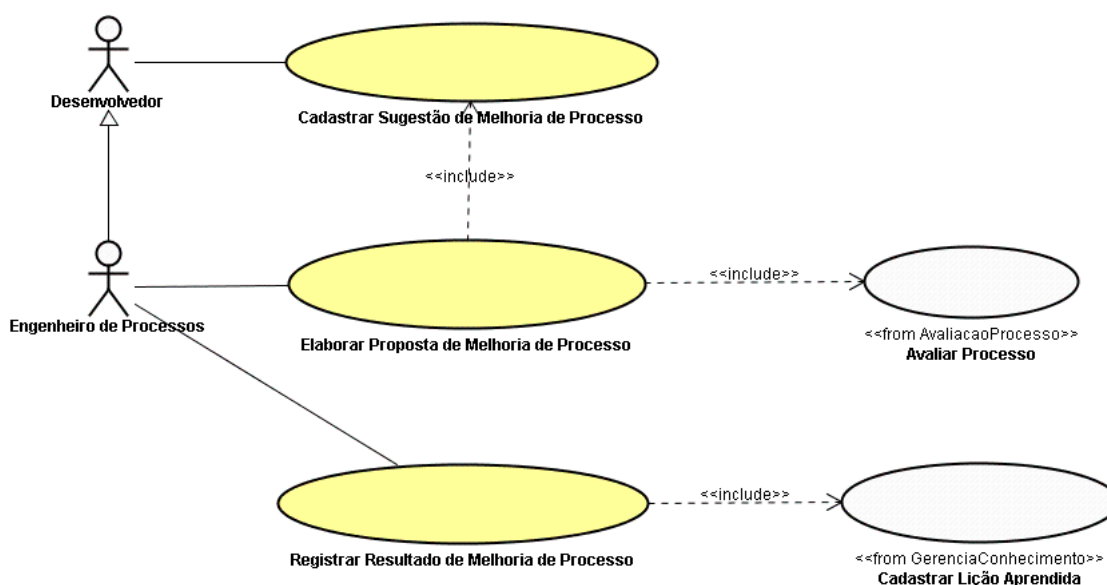


Figura 5.13 – Diagrama de Casos de Uso de MelhoriaODE.

Já em relação aos casos de uso, há três deles que são específicos da ferramenta MelhoriaODE, a saber *Cadastrar Sugestão de Melhoria de Processo*, *Elaborar Proposta de Melhoria de Processo* e *Registrar Resultado de Melhoria de Processo*. Os demais casos de uso são referentes a outras ferramentas de ODE, sendo *Avaliar Processo* referente à ferramenta AvaliaODE, que está sendo utilizado neste contexto para a consulta de observações de avaliação de um processo, e *Cadastrar Lição Aprendida*, referente ao sistema de Gerência de Conhecimento de ODE, que está sendo utilizado aqui para o registro de lições aprendidas com a melhoria de um processo. A seguir, são descritos sucintamente os casos de uso específicos da ferramenta MelhoriaODE.

- **Cadastrar Sugestão de Melhoria de Processo:** Este caso de uso permite que desenvolvedores façam sugestões de melhoria para um processo. Seu objetivo é permitir que qualquer usuário do ambiente ODE possa incluir sugestões de

melhoria em processos. Essas sugestões podem ser referentes a uma inclusão, alteração ou exclusão de um processo ou de uma atividade de um processo.

- **Elaborar Proposta de Melhoria de Processo:** Este caso de uso permite que o engenheiro de processos, analisando sugestões de melhoria feitas pelos desenvolvedores, elabore propostas de melhoria para um processo. Seu objetivo é permitir a seleção de sugestões de melhoria que farão parte da proposta de melhoria, sendo que, para tal, o engenheiro de processos pode consultar as observações de avaliação do processo antes de finalizar a proposta de melhoria. Além disso, o engenheiro de processos pode também avaliar as sugestões de melhoria, tendo a opção de selecionar as que serão contempladas na proposta, as que deverão ser reavaliadas em outra ocasião (opção padrão), e as que devem ser descartadas de modo a não serem mais apresentadas como sugestão.
- **Registrar Resultado de Melhoria de Processo:** Este caso de uso permite que o engenheiro de processos registre os resultados alcançados com uma melhoria de processo. Além do registro dos resultados, é possível registrar lições aprendidas com a melhoria de processo em questão. Por fim, o engenheiro de processos pode finalizar uma melhoria de processo, fato esse que impede que novos resultados sejam incluídos ou excluídos.

#### 5.4.2 – Modelo Conceitual de MelhoriaODE

Uma vez definidos os casos de uso da ferramenta, deve-se elaborar o modelo conceitual para tratá-los. A figura 5.14 apresenta o diagrama de pacotes da ferramenta MelhoriaODE.

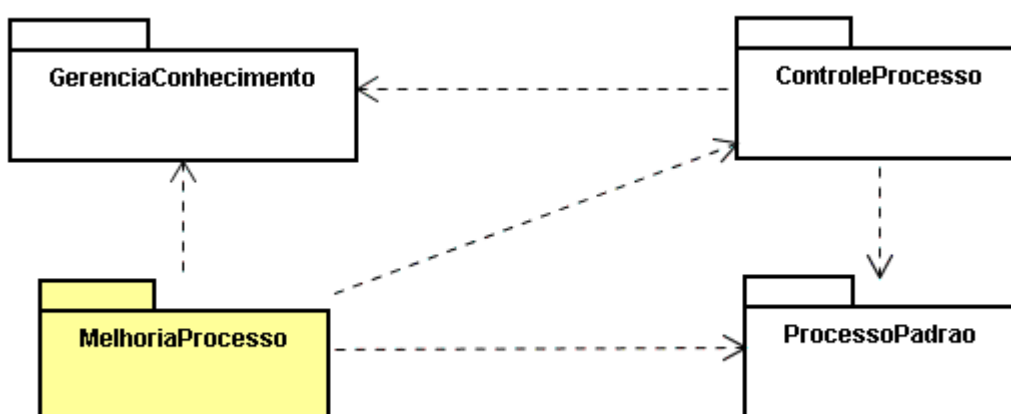


Figura 5.14 – Diagrama de Pacotes de MelhoriaODE.

Conforme discutido na Seção 5.3, o pacote *ProcessoPadrao* contém as classes relacionadas com a definição dos processos padrão de uma organização. Esses processos servem de base para a definição de processos para projetos específicos, que consideram as características particulares do projeto (pacote *ControleProcesso*). Já o pacote *GerenciaConhecimento* contém as classes relacionadas com o sistema de Gerência de Conhecimento de ODE. Por fim, o pacote *MelhoriaProcesso* trata das funcionalidades mais diretamente relacionadas com a melhoria de processos.

A Figura 5.15 apresenta o diagrama de classes do pacote *MelhoriaProcesso*, sendo as suas classes apresentadas em destaque, tendo sido inseridas como contribuição deste trabalho.

Uma das funcionalidades de MelhoriaODE é o cadastro de sugestões de melhoria para um processo padrão, sendo a classe *SugestaoMelhoria* a responsável por tratar as sugestões de melhoria feitas pelos desenvolvedores (classe *RecursoHumano* do pacote *ControleProcesso*). Uma sugestão de melhoria pode ser uma inclusão (classe *SugestaoInclusaoProcesso*), uma exclusão (classe *SugestaoExclusaoProcesso*) ou uma alteração (classe *SugestaoAlteracaoProcesso*) de um processo.

Se uma sugestão de melhoria for do tipo alteração do processo, essa alteração se refere a alterações em suas atividades, tratadas pela classe *SugestaoAlteracaoAtividadeProcesso*. Desse modo, uma vez que o desenvolvedor indique que sua sugestão de melhoria é uma alteração do processo, ele também deve indicar se é uma inclusão (classe *SugestaoInclusaoAtividade*), uma exclusão (classe *SugestaoExclusaoAtividade*) ou uma alteração (classe *SugestaoAlteracaoAtividade*) de uma atividade do processo em questão.

Ao se criar uma proposta de melhoria (classe *PropostaMelhoria*), é criado também um projeto de melhoria (classe *ProjetoMelhoriaProcesso*) para o processo a ser melhorado (classe *ProcessoPadrao*). Uma proposta de melhoria pode conter mais de uma sugestão de melhoria, sendo que todas elas devem ser referentes ao mesmo processo padrão.

Quando esse projeto é executado, podem-se registrar resultados da melhoria do processo (classe *ResultadoMelhoria*) ou, ainda, lições aprendidas (classe *LicaoAprendida* do pacote *GerenciaConhecimento*) com a melhoria do processo.

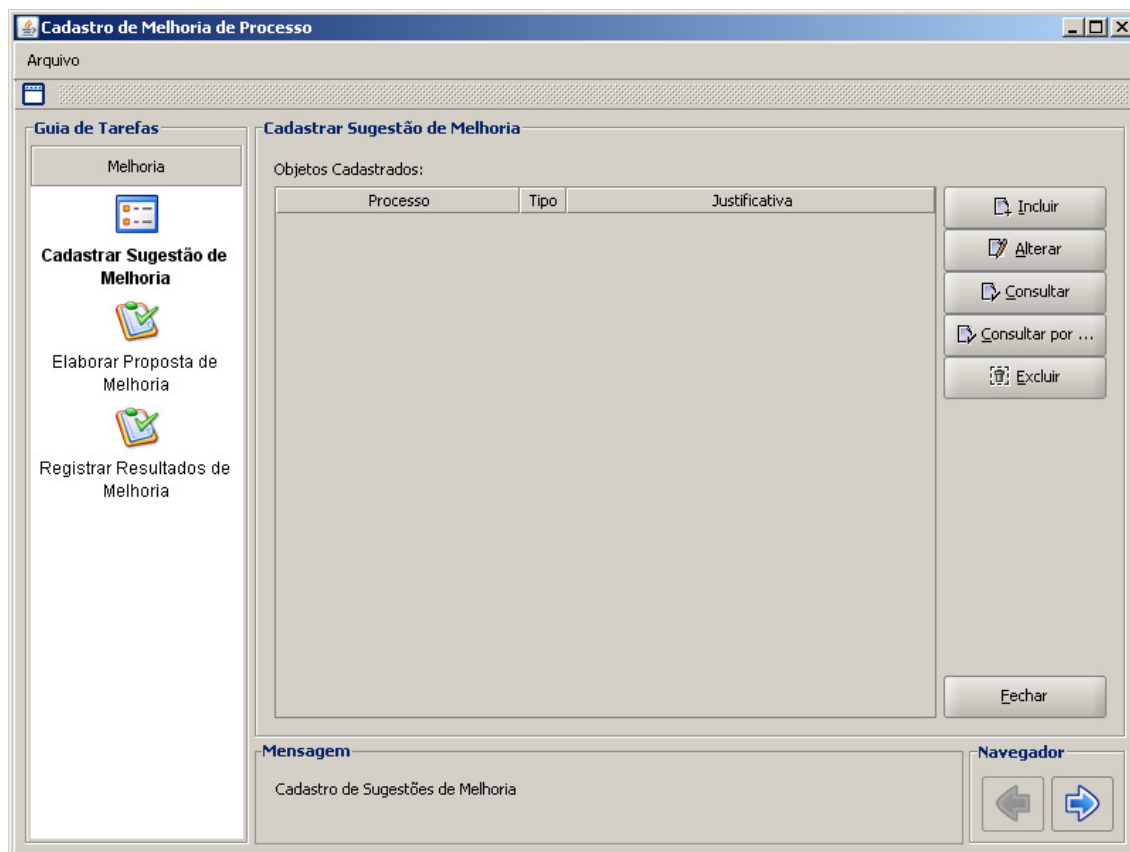




### 5.4.3 – Apresentação de MelhoriaODE

Nesta seção são apresentadas as principais funcionalidades da ferramenta MelhoriaODE, por meio da apresentação de algumas de suas interfaces.

A Figura 5.16 apresenta a tela principal da ferramenta MelhoriaODE, seguindo os padrões de interface do ambiente ODE.



**Figura 5.16 – Tela principal de MelhoriaODE.**

Uma vez selecionado o item de menu *Cadastrar Sugestão de Melhoria*, é permitido ao usuário incluir, alterar, consultar ou excluir uma sugestão de melhoria, bem como a consulta de sugestões de melhoria por processo. Para ilustrar a utilização da ferramenta, a Figura 5.17 mostra a inclusão de uma sugestão de melhoria. O desenvolvedor deve indicar primeiramente a natureza da sugestão de melhoria. No exemplo da Figura 5.17, foi selecionada a opção de alterar um sub-processo. Nesse momento é solicitado que sejam identificados o processo padrão e o sub-processo padrão que devem ser alterados. A seguir uma justificativa acerca da sugestão de melhoria deve ser informada. Por fim, deve ser indicado o tipo da alteração e, dependendo do tipo selecionado, devem-se informar outras informações acerca da sugestão. A Figura 5.18 ilustra uma alteração de atividade de processo.

**Dados da Sugestão de Melhoria**

Alteração

Natureza da sugestão de melhoria:

☐ Incluir um sub-processo

☒ Alterar um sub-processo

☐ Excluir um sub-processo

Processo Padrão: Processo Padrão LabES

Sub-Processo Padrão: Gerência de Projeto

Justificativa

A atividade de acompanhamento do projeto precisa ser decomposta em sub-atividades, pois atualmente está em nível macro.

Tipo da Alteração:

Alterar atividade

Tipo	Atividade
A	Acompanhamento de Projeto

Incluir

Alterar

Consultar

Excluir

OK

Fechar

**Figura 5.17 – Inclusão de Sugestão de Melhoria em MelhoriaODE.**

**Alteração de Atividade**

Atividade a ser alterada: Acompanhamento de Projeto

Descrição:

Decompor a atividade Acompanhamento de Projeto em sub-atividades para facilitar o acompanhamento dos marcos do projeto.

Ok

Fechar

**Figura 5.18 – Inclusão de Sugestão de Alteração de Atividade em MelhoriaODE.**

Com base em sugestões de melhoria previamente cadastradas, o engenheiro de processos pode, então, elaborar uma proposta de melhoria. A Figura 5.19 ilustra a realização dessa atividade em MelhoriaODE. Em primeiro lugar deve ser selecionado o processo ao qual se refere a proposta de melhoria. A seguir devem ser descritos os objetivos, os benefícios e os problemas que se espera sanar com a proposta de melhoria.

**Dados da Proposta de Melhoria**

Dados Gerais | Sugestões de Melhoria

Proposta de Melhoria para um: ☐ Processo Padrão ☒ Sub-processo

Processo Padrão:

Sub-Processo Padrão:

...

Descrição sucinta:

Objetivos:

Benefícios:

Problemas a serem sanados:

OK Fechar

**Figura 5.19 – Dados Gerais de uma Proposta de Melhoria em MelhoriaODE.**

Além das informações gerais, o engenheiro de processo deve também selecionar quais das sugestões de melhoria para o processo em questão ele quer considerar como

parte do escopo dessa proposta de melhoria. Selecionando a aba *Sugestões de Melhoria*, é apresentada para o engenheiro de processos uma lista com todas as sugestões de melhoria ainda não avaliadas para o processo em questão. Nesse momento, conforme ilustra a Figura 5.20, ele pode selecionar quais sugestões farão parte da proposta, bem como quais devem ser descartadas ou, então, que devem ser reavaliadas em outra oportunidade.

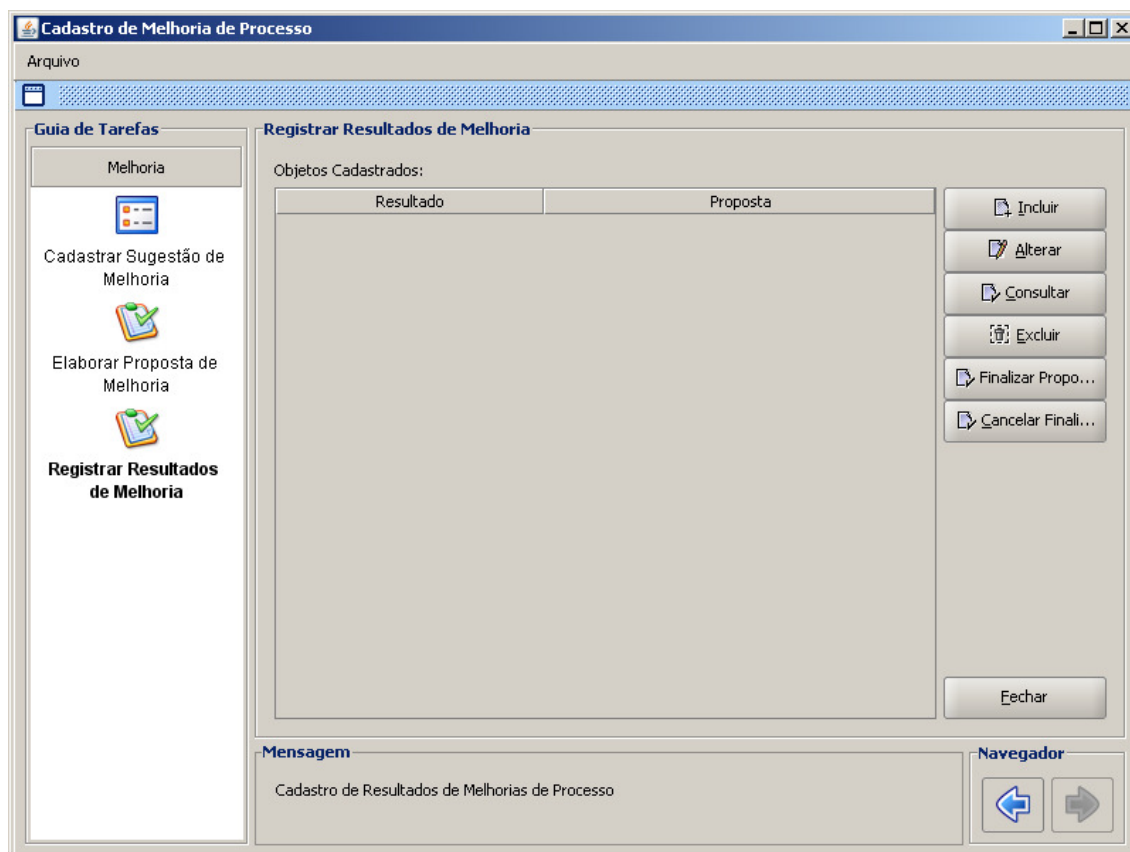
A janela "Dados da Proposta de Melhoria" possui duas abas: "Dados Gerais" e "Sugestões de Melhoria". A aba "Sugestões de Melhoria" está ativa e exibe a seguinte tabela:

Processo	Justificativa	Selecionar	Descartar	Reavaliar
Gerência de Projeto	A atividade de acompanhamento ...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gerência de Projeto	Incluir nova atividade para elabor...	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Abaixo da tabela, há um botão "Consultar". Na base da janela, há dois botões: "OK" e "Fechar".

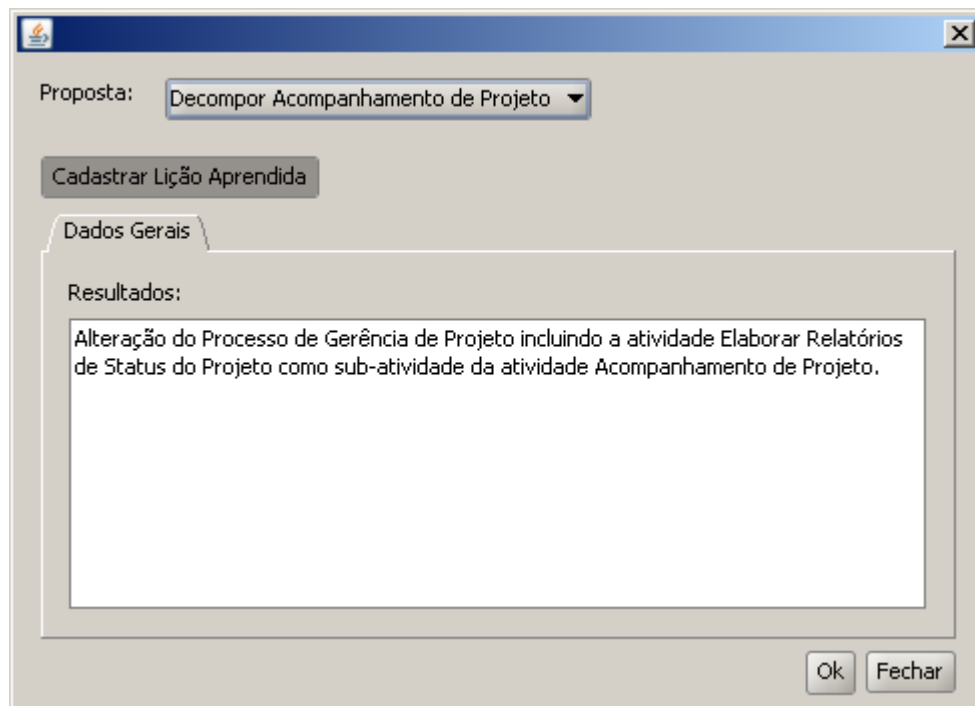
**Figura 5.20 – Seleção de Sugestões de Melhoria para uma Proposta de Melhoria em MelhoriaODE.**

Por fim, o engenheiro de processos pode registrar resultados obtidos com a melhoria do processo, como ilustra a Figura 5.21. Por meio desta funcionalidade, o engenheiro de processos pode incluir, alterar, consultar ou excluir um resultado de melhoria, bem como finalizar ou cancelar a proposta de melhoria de processo. A finalização da proposta somente é possível quando houver pelo menos um resultado registrado para ela. Uma vez que a proposta seja finalizada, não é mais permitido que sejam alterados seus dados.



**Figura 5.21 – Registro de Resultados para uma Proposta de Melhoria em MelhoriaODE.**

A Figura 5.22 ilustra a inclusão de um resultado de melhoria. Para incluir um novo resultado, deve-se selecionar a proposta para qual o resultado será registrado e descrever textualmente o resultado em si. Além disso, também podem ser incluídas lições aprendidas com a melhoria de processo, utilizando-se funcionalidades da infraestrutura de Gerência de Conhecimento de ODE.



**Figura 5.22 – Inclusão de um Resultado para uma Proposta de Melhoria em MelhoriaODE.**

### ***5.5 Considerações Finais do Capítulo***

Neste capítulo foram apresentados o ambiente ODE e as ferramentas AvaliaODE, responsável pela avaliação de processos e produtos de software, e MelhoriaODE, responsável pela melhoria de processos de software, desenvolvidas no contexto deste trabalho para prover apoio automatizado, ainda que parcialmente, ao processo de AMP.

### Considerações Finais

O objetivo deste capítulo é tecer comentários finais acerca do trabalho realizado, sobretudo no que se refere às contribuições e perspectivas futuras de pesquisas, levantadas a partir dos resultados e idéias que surgiram ao longo da realização deste trabalho.

#### *6.1 Conclusões*

O fato das organizações de software perceberem que a qualidade do produto é, em geral, uma consequência da qualidade dos processos adotados no seu desenvolvimento faz com que elas se preocupem e valorizem a melhoria contínua de seus processos organizacionais. Essa percepção corrobora a importância da avaliação e melhoria de processos de software.

No decorrer do desenvolvimento deste trabalho houve uma dedicação ao estudo de algumas áreas importantes da Engenharia de Software, em especial a Avaliação e Melhoria de Processos de Software (AMP), foco principal deste trabalho. Entretanto, devido a sua complexidade, amplitude e inter-relação com outras áreas, foram estudadas também as áreas de Processo de Software, incluindo normas e modelos de qualidade de processo, Medição, Ontologias e Ambientes de Desenvolvimento de Software (ADSs).

Dado que um dos objetivos deste trabalho foi prover apoio automatizado ao processo de AMP, fez-se um esforço inicial para a definição de um processo de AMP que fosse baseado no modelo de qualidade MPS.BR (SOFTEX, 2007a). A escolha desse modelo de qualidade como referência levou em conta os fatos de ele ser reconhecido nacionalmente no mercado de software e, ainda, ser baseado em normas e modelos internacionais de qualidade, a saber: CMMI (SEI, 2006a), ISO/IEC 12207 (ISO/IEC, 2008) e ISO/IEC 15504 (ISO/IEC, 2003).

Para a definição do processo foram estudadas, além da área de Avaliação e Melhoria de Processos de Software, as áreas de Processo de Software e Medição. O estudo da área de Processo de Software foi natural, uma vez que o objetivo era a definição de um processo de AMP e esse processo possui uma grande interação com outros processos do ciclo de vida de software, com destaque para o processo de Medição, que pode ser visto como uma base para a avaliação de um processo. Logo, seu

estudo também foi essencial para o desenvolvimento do trabalho. Por fim, diversos modelos e normas de qualidade de processo, dentre eles a ISO/IEC 12207, a ISO/IEC 15504, o CMMI e o MPS.BR, foram estudados para que o processo definido fosse baseado nas boas práticas por eles sugeridas.

Vale ressaltar que o objetivo principal do trabalho foi estabelecer uma conceituação base para o domínio de AMP a ser usada como base para o desenvolvimento de ferramentas de apoio ao processo de AMP. Entretanto, conforme já citado anteriormente, o processo de AMP possui uma interação considerável com outros processos do ciclo de vida de software e, sendo assim, outra premissa importante levada em consideração foi a integração das ferramentas de apoio à AMP com outras que apóiam a automatização de outros processos. Isso foi facilitado pelo fato das ferramentas terem sido desenvolvidas integradas a ODE (*Ontology-based software Development Environment*) (FALBO et al., 2003), um Ambiente de Desenvolvimento de Software (ADS) centrado em processo e baseado em ontologias.

Uma vez que ODE é um ADS baseado em ontologias, o estudo da área de Ontologias também foi necessário. Esse estudo em conjunto ao estudo de ODE teve como resultado a decisão do desenvolvimento de uma Ontologia de Qualidade de Software com foco no domínio de AMP. Anteriormente existia no âmbito do Projeto ODE uma ontologia de qualidade que enfocava principalmente a avaliação de produtos de software. O objetivo da nova ontologia de qualidade de software foi abranger a avaliação e melhoria de processos de software, contemplando também a avaliação de produtos de software.

As ferramentas desenvolvidas no contexto deste trabalho foram AvaliaODE, responsável pelo apoio à avaliação de processos e produtos de software e MelhoriaODE, responsável pelo apoio à melhoria de processos de software. Por estarem inseridas no contexto de ODE, foi possível explorar a questão da integração entre as ferramentas existentes e as desenvolvidas, como por exemplo, a interação entre a ferramenta AvaliaODE e a ferramenta GQA-ODE (PALACIO, 2007), que apóia o processo de Garantia de Qualidade e compartilha com Avalia-ODE algumas de suas funcionalidades, como por exemplo, o cadastro e resposta a *checklists*, utilizados em Avalia-ODE como critérios para a avaliação de um processo ou produto de software.



Desse modo, as principais contribuições deste trabalho foram:

- Desenvolvimento de uma Ontologia de Qualidade de Software com foco no domínio de Avaliação e Melhoria de Processos de Software, incluindo, ainda, conceitos relativos à Medição.
- Definição de um processo de AMP, baseado no modelo de qualidade MPS.BR, que serviu de base para a identificação de funcionalidades de apoio a serem providas por ferramentas desenvolvidas e integradas a ODE.
- Desenvolvimento de ferramentas que apóiam parcialmente o processo de AMP definido em ODE, a saber: AvaliaODE, que apóia parcialmente a avaliação de processo e produtos de software, e MelhoriaODE, que provê funcionalidades básicas de apoio à melhoria de processos de software.

Fazendo uma analogia à própria ontologia de qualidade de software desenvolvida, mais especificamente utilizando os conceitos da sub-ontologia de avaliação, é importante refletir sobre quais são os pontos positivos, os pontos negativos e as oportunidades de melhoria deste trabalho.

O primeiro ponto positivo que merece destaque foi a decisão de se definir um processo de AMP, mesmo que de modo geral abrangendo apenas atividades, antes de pensar em prover apoio automatizado para ele. A definição desse processo auxiliou muito a identificação de requisitos que deveriam ser tratados nas ferramentas de apoio à AMP. Ainda em relação à definição do processo de AMP, outro ponto positivo foi baseá-lo no modelo de qualidade MPS.BR. Sabe-se que o MPS.BR tem ganhado, cada vez mais, o reconhecimento do mercado nacional de software. Além disso, ele é baseado em normas e modelos de qualidade internacionais. Desse modo, definir um processo que tenha como base as boas práticas advogadas pelo MPS.BR, respalda, de certa forma, que o processo definido tenha credibilidade, mesmo sem ter sido utilizado em casos práticos.

Um ponto negativo que pode ser citado em relação ao processo de AMP definido é exatamente a falta de sua utilização na prática. Conforme discutido ao longo deste trabalho, um processo deve ser utilizado em um projeto piloto, avaliado, ajustado e institucionalizado, para depois ser alvo da melhoria contínua. Assim, sabe-se que apenas o passo inicial foi dado. Devido a restrições de tempo não foi viável a sua utilização em empresas de software para que o mesmo fosse validado em casos reais e, portanto, a sua utilização em casos reais fica como perspectiva de trabalhos futuros.

Já em relação a oportunidades de melhoria, um ponto que merece destaque é a definição completa do processo de AMP, contemplando além das atividades atuais o seu desmembramento em outros níveis de sub-atividades, caso um estudo mais aprofundado indique a necessidade de uma granularidade mais fina das atividades, além da definição de artefatos, recursos e procedimentos para todas as atividades do processo..

Em relação ao desenvolvimento da ontologia de qualidade de software, o primeiro ponto forte que merece destaque foi a preocupação em sua aplicabilidade tanto para avaliação de processos quanto de produtos de software. Nos estudos e pesquisas realizados durante o desenvolvimento dessa ontologia um fato que chamou a atenção foi a dificuldade de se encontrar trabalhos que relatassem a avaliação de processos e produtos de software de maneira integrada. Na literatura muito se fala de avaliação de produto ou avaliação de processo, mas não da avaliação conjunta deles. Assim, o primeiro desafio da ontologia desenvolvida foi capturar a conceituação envolvida tanto na avaliação de processos quanto de produtos de software, o que pode ser comprovado pela instanciación discutida no capítulo 4 desta dissertação. Outro ponto forte que merece ser destacado é a inclusão do domínio de medição na ontologia de qualidade de software. Apesar de não ser o tema foco deste trabalho, percebeu-se que, devido à forte inter-relação entre os processos de AMP e Medição, seria de muita valia a inclusão de conceitos de medição na ontologia. No decorrer do desenvolvimento essa percepção inicial se comprovou de tal forma que medição deu origem a uma sub-ontologia. É fato que para melhorar é preciso avaliar e, para avaliar é preciso medir. Logo, a inclusão de conceitos de medição no desenvolvimento da ontologia de qualidade a torna mais abrangente e potencialmente útil.

Como ponto negativo da ontologia de qualidade de software desenvolvida pode-se citar o fato dela não ter sido avaliada segundo uma ontologia de fundamentação (GUIZZARDI, 2005), o que pode apontar problemas na conceituação proposta. Essa avaliação seria importante para confirmar se a conceituação proposta está aderente a conceituações mais genéricas que são subjacentes ao domínio considerado, neste caso, a Avaliação e Melhoria de Processos de Software, incluindo Medição.

Já uma oportunidade de melhoria identificada para a ontologia seria a realização de ajustes em sua conceituação para atender também a níveis mais altos de maturidade, tais como os níveis A e B do modelo de qualidade MPS.BR. Uma vez que tais níveis sejam considerados, é muito provável que a ontologia precise ser ajustada. Esse ajuste é previsível, uma vez que em níveis mais altos de maturidade há uma preocupação com a

gerência quantitativa do processo, fato esse que implica no controle estatístico do desempenho dos processos. Segundo BARCELLOS e ROCHA (2008), um dos principais problemas em organizações que almejam tais níveis é a falta de uma avaliação das medidas / bases de medidas de uma organização que realmente sejam úteis para tal controle. Essa questão não foi contemplada e é possível que ela leve a conceitos e relações não capturados pela ontologia proposta.

Em relação às ferramentas AvaliaODE e MelhoriaODE, os pontos positivos que merecem destaque são: (i) estão integradas ao ambiente de desenvolvimento de software ODE, compartilhando inclusive funcionalidades com outras ferramentas do ambiente; (ii) foram desenvolvidas com base nos requisitos identificados no processo de AMP definido contribuindo, assim, para sua aderência, mesmo que parcialmente, às boas práticas ditadas pelo MPS.BR; e (iii) foram desenvolvidas com base no modelo conceitual proposto pela ontologia de qualidade de software, favorecendo a interoperabilidade entre as mesmas e facilitando sua integração a outras ferramentas de ODE, como GQA-ODE (PALACIO, 2007).

Assim como ocorreu com o processo de AMP definido, um ponto negativo das ferramentas desenvolvidas foi a falta de sua aplicação em casos práticos dentro de uma organização de software. Essa aplicação seria interessante para validar se as funcionalidades presentes em AvaliaODE e MelhoriaODE realmente apóiam o processo de Avaliação e Melhoria de Processos de Software de uma organização. Além disso, essa aplicação também serviria para que fossem identificadas possíveis falhas e fossem sugeridas melhorias para as ferramentas desenvolvidas, buscando-se, com isso, uma melhoria contínua também em relação às funcionalidades das ferramentas.

Como oportunidades de melhoria para as ferramentas, vale ressaltar que nem todas as funcionalidades identificadas para a automatização do processo de AMP foram desenvolvidas no contexto deste trabalho. Desse modo, a implementação das demais funcionalidades tanto em AvaliaODE quanto em MelhoriaODE, e quem sabe até mesmo em outras ferramentas, como por exemplo, uma ferramenta específica para o domínio de Medição, pode ser apontada como bons exemplos de oportunidades de melhoria referentes às ferramentas desenvolvidas.

Em suma, de maneira geral, a principal contribuição deste trabalho foi trabalhar uma conceituação do domínio de AMP de forma ampla e seu uso para o desenvolvimento de ferramentas de apoio a um processo de AMP definido.

## ***6.2 Perspectivas Futuras***

Conforme citado na seção anterior, há muitas oportunidades de melhoria e pontos fracos deste trabalho que podem ser considerados em trabalhos e pesquisas futuros, dentre eles:

- Definição completa do processo de AMP, considerando a necessidade da definição de artefatos, recursos e procedimentos para as atividades do processo.
- Utilização do processo de AMP definido em casos reais, sendo utilizado primeiramente em um projeto piloto, para ser avaliado, ajustado e institucionalizado, e em seguida entrar no ciclo da melhoria contínua.
- Ajustes na ontologia de qualidade de software dando atenção especial às exigências da gerência quantitativa do processo.
- Avaliação da ontologia de qualidade de software segundo uma ontologia de fundamentação com o objetivo de confirmar se a conceituação proposta está aderente a conceituações mais genéricas que são subjacentes ao domínio de AMP.
- Implementação das demais funcionalidades identificadas, seja como acréscimo nas próprias ferramentas AvaliaODE e MelhoriaODE, seja no desenvolvimento de outras ferramentas, como é o caso especial das funcionalidades referentes à medição, para apoiar o processo de AMP por completo.
- Utilização das ferramentas desenvolvidas em casos reais, com o intuito de validar se as funcionalidades propostas realmente apóiam o processo de AMP e a partir de seu uso, evolução das mesmas.

Além das perspectivas mencionadas acima, outras ainda podem ser consideradas em trabalhos futuros, sendo algumas destacadas a seguir.

Da mesma maneira que o processo de AMP e a ontologia de qualidade podem sofrer mudanças ao se analisar níveis mais altos de maturidade do MPS.BR, é provável que o conjunto de funcionalidades de apoio levantadas e, por conseguinte, as ferramentas AvaliaODE e MelhoriaODE também precisem evoluir. Logo, elas devem ser alvo de aperfeiçoamento com vistas a tratar aspectos desses níveis. Essa evolução é

natural, na medida em que as funcionalidades e a estrutura das ferramentas são baseadas tanto no processo de AMP quanto nos conceitos na ontologia de qualidade.

Analisando as interações entre o processo de AMP e outros processos do ciclo de vida de um software, discutidas no capítulo 3 desta dissertação, percebe-se uma ligação forte entre o processo de AMP e o processo de Garantia da Qualidade. Tanto isso é verdade que algumas funcionalidades da ferramenta de garantia da qualidade de ODE, GQA-ODE (PALÁCIO, 2007), foram alteradas como resultado deste trabalho, para torná-la aderente à ontologia de qualidade. Porém, apenas as funcionalidades diretamente envolvidas com a AMP foram analisadas e uma perspectiva de trabalho futuro seria analisar de forma mais completa GQA-ODE sob o ponto de vista da ontologia de qualidade de software desenvolvida neste trabalho, com o intuito de torná-la aderente à ontologia ou, quem sabe, até mesmo evoluir a própria ontologia de qualidade para que ela contemple conceitos do domínio de garantia de qualidade que porventura ela ainda não tenha contemplado.

Ao analisar essas questões para o processo de Garantia da Qualidade, outro ponto interessante de pesquisa seria realizar o mesmo estudo para os processos de Verificação e Validação. Novamente analisando as inter-relações entre o processo de AMP e os processos de Verificação e Validação, percebe-se que ambos possuem interações análogas ao existente entre os processos de AMP e de Garantia da Qualidade. Assim, de maneira análoga ao processo de Garantia da Qualidade, além de se pensar em evoluir a ontologia de qualidade para que ela contemple esses dois processos, vale também pensar em desenvolver ferramentas de apoio aos processos de Verificação e Validação para ODE.

Outra abordagem interessante, pouco explorada no contexto deste trabalho, é a utilização da Gerência de Conhecimento para apoiar o processo de AMP. Apesar até de haver um relacionamento pequeno entre a ferramenta MelhoriaODE e o sistema de Gerência de Conhecimento de ODE, quando durante a inclusão de resultados de uma melhoria podem, também, ser incluídas lições aprendidas com essa melhoria, isso é muito pouco pelo potencial de informações importantes que a Gerência de Conhecimento pode prover ao Engenheiro de Processos. Apenas como um exemplo, poderia haver funcionalidades em ODE, acessíveis a partir de AvaliaODE e MelhoriaODE, em que o Engenheiro de Processos pudesse ter acesso a experiências passadas em melhoria de processos, antes de definir a proposta de uma nova melhoria.

Um fato importante que merece destaque como perspectiva futura de trabalho é a necessidade de um estudo mais amplo dos impactos causados pela reestruturação da ontologia de qualidade de software em outras ontologias de ODE que reutilizam parcialmente a sua conceituação. Por exemplo, a ontologia de requisitos de software (NARDI, 2006) faz referência a conceitos da ontologia de qualidade. Esse estudo pode inclusive apontar necessidade de ajustes em ferramentas derivadas dessas ontologias, tal como ReqODE (MARTINS et al., 2006). De maneira geral, um estudo sobre evolução de ontologias e impactos em sistemas derivados delas é bastante relevante no contexto do Projeto ODE, tendo em vista que o mesmo adota a premissa de construção de ferramentas tomando por base ontologias.

Enfim, como pode ser observado pelas reflexões acima, ainda que um considerável trabalho tenha sido realizado ao longo desta pesquisa, muitos outros podem ser desenvolvidos a partir das idéias apresentadas neste trabalho, ficando, assim, como perspectivas para novos estudos e pesquisas futuras.

## Referências Bibliográficas

- ABNT (2000) NBR ISO 9000 – *Sistemas de gestão da qualidade – Fundamentos e vocabulário*.
- AHONEN, J. J., FORSELL, M., TASKINEN, S., (2002) “A Model but Practical Software Process Modeling Technique for Software Process Improvement”, *Software Process: Improvement and Practice*, 7:33-44.
- ALBUQUERQUE, A. B., (2005) “Melhoria de Processos nos Ambientes Configurados TABA”, Exame de Qualificação ao Doutorado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.
- ANDRADE, J. M., ALBUQUERQUE, A. B., CAMPOS, F. B., ROCHA, A. R. (2004) “Consequências e Características de um Processo de Desenvolvimento de Software de Qualidade e Aspectos que o Influenciam: Uma Avaliação de Especialistas”, In: *Anais do III Simpósio Brasileiro de Qualidade de Software, SBQS’2004*, Brasília, Brasil. Anais... 2004, p. 79-93.
- ANDRADE, J. M. S., (2005) “Avaliação de Processos de Software em Ambientes de Desenvolvimento de Software Orientados à Organização”, *Dissertação de Mestrado*, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.
- ARANTES, L. O., FALBO, R. A., GUIZZARDI, G., (2007) “Evolving a Software Configuration Management”, *Second Workshop on Ontologies and Metamodeling Software and Data Engineering – WOMSDE’2007*, XXI Simpósio Brasileiro de Engenharia de Software – SBES’2007, João Pessoa.
- BARCELLOS, M. P, ROCHA, A. R. C., (2008) “Avaliação de Bases de Medidas considerando sua Aplicabilidade ao Controle Estatístico de Processos de Software”. In: *VII Simpósio Brasileiro de Qualidade de Software, SBQS 2008*, Florianópolis, Brasil. Anais... 2008, p. 75-90.
- BERGER, P., (2003) “Instanciação de Processos de Software em Ambientes Configurados na Estação TABA”. *Dissertação de Mestrado*, COPPE/UFRJ, Rio de Janeiro.
- BERTOIA, et al. (2006) “An Ontology for Software Measurement”, In: (Calero et al., 2006).
- BERTOLLO, G., FALBO, R. A., (2003) “Apoio Automatizado à Definição de Processos de Software em Níveis”. In: *II Simpósio Brasileiro de Qualidade de Software, SBQS 2003*, Fortaleza, Brasil. Anais... 2003, p. 1-6.
- BERTOLLO, G., (2006) “Definição de Processos em um Ambiente de Desenvolvimento de Software”, *Dissertação de Mestrado*, Mestrado em Informática, UFES, 2006.
- BERTOLLO, G., SEGRINI, B. M., FALBO, R. A., (2006) “Definição de Processos de Software em um Ambiente de Desenvolvimento de Software Baseado em Ontologias”. *V Simpósio Brasileiro de Qualidade de Software*. Vila Velha, Brasil, Maio/Junho 2006. p. 72-86
- BRINGUENTE, A. C., (2008) “Apoio Automatizado à Avaliação e Melhoria de Processos em ODE”, *Monografia de conclusão do curso de Ciência da Computação*, UFES, 2008 (Em Elaboração).

- CALERO, C., RUIZ, F., PIATTINI, M., (eds) (2006) *Ontologies for Software Engineering and Software Technology*, Springer-Verlag Berlin Heidelberg.
- CAMPOS, F. B., (2005) Melhoria de Processos e Evolução do Meta-Ambiente TABA, Exame de Qualificação ao Doutorado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.
- CARVALHO, V. A., ARANTES, L. O., FALBO R. A., (2006) "EstimaODE: Apoio a Estimativas de Tamanho e Esforço no Ambiente de Desenvolvimento de Software ODE". Anais do V Simpósio Brasileiro de Qualidade de Software, p. 12-26, Vila Velha, Brasil, Maio 2006.
- CATTANEO, F., FUGGETA, A., SCIUTO, D., (2001) Pursuing Coherence in Software Process Assessment and Improvement , Software Process: Improvement and Practice, 6:3-22.
- CHRISSIS, M. B., KONRAD, M., SHRUM, S., (2003) *CMMI: Guidelines for Process Integration and Product Improvement*, Addison Wesley.
- CONRADI, R., FUGGETTA, A., (2002) "Improving Software Process Improvement", IEEE Software, July/August.
- COTA, R. I.; MENEZES, C. S.; FALBO, R. A. (2004) "Modelagem Organizacional Utilizando Ontologias e Padrões de Análise". In: VII Workshop Iberoamericano de Ingeniería de Requisitos y Desarrollo de Ambientes de Software - IDEAS'2004, pp. 56-67, Arequipa, Perú, Maio 2004.
- DEVEDZIC, V., (1999) "Ontologies: Borrowing from Software Patterns". ACM intelligence Magazine, Fall 1999, pp. 14-24.
- DUARTE, K. C., FALBO, R. A., (2000) "Uma Ontologia de Qualidade de Software", Anais do VII Workshop de Qualidade de Software, XIV Simpósio Brasileiro de Engenharia de Software, pp. 275-285, João Pessoa, Brasil.
- FABBRINI, F., FANTINI, E., FUSANI, M., LAMI, G., (2003) "Performing SPICE Assessments: Yet Another Tool". In Joint ESA - 3rd International SPICE Conference on Process Assessment and Improvement 17-21 March 2003 ESTEC, Noordwijk, The Netherlands.
- FALBO, R. A., NATALI, A. C. C., MIAN, P. G., BERTOLLO, G., RUY, F. B., (2003) "ODE: Ontology-based software Development Environment", In: Memórias de IX Congreso Argentino de Ciencias de la Computación, p. 1124-1135, La Plata, Argentina.
- FALBO, R. A., (2004) "Experiences in Using a Method for Building Domain Ontologies" Proc. of the 16th International Conference on Software Engineering and Knowledge Engineering, International Workshop on Ontology In Action, Banff, Canada.
- FALBO, R. A., RUY, F. B., BERTOLLO, G., TOGNERI, D. F., (2004a) "Learning How to Manage Risks Using Organizational Knowledge". Advances in Learning Software Organizations (Proceedings of the 6th International Workshop on Learning Software Organizations - LSO'2004), Melnik G. and Holz, H. (Eds.): LNCS 3096, pp. 7-18, Springer-Verlag Berlin Heidelberg, Banff, Canada, June 2004.
- FALBO, R. A., ARANTES, D. O., NATALI, A. C. C., (2004b) "Integrating Knowledge Management and Groupware in a Software Development Environment". In: 5th



- International Conference on Practical Aspects of Knowledge Management - PAKM'2004, Vienna, Springer-Verlag Berlin Heidelberg, Vol. 3336, pp. 94-105.
- FALBO, R. A., BERTOLLO, G., (2005) "Establishing a Common Vocabulary for Software Organizations Understand Software Processes", International Workshop on Vocabularies, Ontologies and Rules for the Enterprise, Enschede, The Netherlands.
- FLORAC, W., CARLETON, A. E., (2000) *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, Addison-Wesley.
- FUGGETTA, A., (2000) "Software Process: A Roadmap", In: Proceedings of The Future of Software Engineering, ICSE'2000, Limerick, Ireland.
- GRUHN, V. (2002) "Process-Centered Software Engineering Environments: A Brief History and Future Challenges", In: Annals of Software Engineering 14, p. 363-382.
- GUARINO, N. (1998) "Formal Ontology in Information System". In: Proceedings of the First Int. Conference on Formal Ontology in Information Systems, Trento, Italy, June 1998. p.3-15.
- GUIZZARDI, G., (2000) "Desenvolvimento para e com Reuso: Um Estudo de Caso no Domínio de Vídeo sob Demanda". Dissertação de Mestrado em Informática, Universidade Federal do Espírito Santo, 2000.
- GUIZZARDI, G., (2005) *Ontological Foundations for Structural Conceptual Models*, Universal Press, The Netherlands.
- HARDGRAVE, B. C., ARMSTRONG, D. J., (2005) "Software Process Improvement: It's a Journey, Not a Destination", Technical Opinion, Communications of the ACM.
- HARRISON, W., OSSHER, H., TARR, P., (2000) "Software Engineering Tools and Environments: A Roadmap". Proceedings of The Future of Software Engineering (ICSE'2000). Limerick, Ireland, p. 263 – 277, 2000.
- HUMPHREY, W., (1989) *Managing the Software Process*, Reading, MA: Addison-Wesley.
- ISO/IEC (1999) ISO/IEC 14598 *Information Technology – Software product evaluation*.
- ISO/IEC (2001), ISO/IEC 9126-1:2001 – *Software Engineering – Product Quality – Part 1: Quality model*.
- ISO/IEC (2002) ISO/IEC 15939:2002 – *Software Engineering – Software Measurement Process*.
- ISO/IEC (2003), ISO/IEC 15504 - *Information Technology – Process Assessment*, ISO/IEC 15504, Geneva, Switzerland: ISO, 2003.
- ISO/IEC (2008), ISO/IEC 12207 - *Information Technology - Software Life Cycle Processes*.
- JASPER, R., USCHOLD, M. (1999) "A Framework for Understanding and Classifying Ontology Applications", Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods, Stockholm, Sweden.
- KAN, S. H., (2003) *Metrics and Models in Software Quality Engineering*, Second Edition, Addison-Wesley.

- MARTIN, M. A., OLSINA, L., (2003) "Towards an Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System", First Latin American Web Congress (LA-WEB'03).
- MARTINS, A. F., NARDI, J. C., FALBO, R. A., (2006) "ReqODE: Uma Ferramenta de Apoio a Engenharia de Requisitos Integrada ao Ambiente ODE". In: ANAIS DA XIII SESSÃO DE FERRAMENTAS. XX Simpósio Brasileiro de Engenharia de Software. Florianópolis-SC, Brasil, pp. 31-36.
- MASON, J., (2003) Aligning Workforce Development & Software Process Improvement Strategy for Accelerated Adoption of Software Engineering Capability, Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03).
- McGARRY, J. et al., (2002) *Practical Software Measurement*, Addison-Wesley.
- MEHNER, T., (1999) Siemens Process Assessment Approach, In: Richard Messnarz and Colin Tully (eds.), *Better Software Practice for Business Benefit: Principles and Experience*, IEEE Computer Society, cap. 8, pp. 199-212.
- MIAN, P. G., (2003) "ODEd: Uma Ferramenta de Apoio ao Desenvolvimento de Ontologias em um Ambiente de Desenvolvimento de Software". Dissertação de Mestrado, Mestrado em Informática, UFES, 2003.
- MIAN, P. G., FALBO, R. A., (2003) "Building Ontologies in a Domain Oriented Software Engineering Environment". IX Congreso Argentino de Ciencias de la Computación, p.930-941 La Plata, Argentina, Outubro 2003.
- MORO, R. D., NARDI, J. C., FALBO, R. A. (2005) "Uma Ferramenta de Acompanhamento de Projetos Integrada a um Ambiente de Desenvolvimento de Software", In: XIX Simpósio Brasileiro de Engenharia de Software, Uberlândia, Brasil.
- NARDI, J. C., (2006) "Apoio de Gerência de Conhecimento à Engenharia de Requisitos em um Ambiente de Desenvolvimento de Software". Dissertação de Mestrado, Mestrado em Informática, UFES, 2006.
- NUNES, B. V., (2005) "Integrando Gerência de Configuração de Software, Documentação e Gerência de Conhecimento em um Ambiente de Desenvolvimento de Software". Dissertação de Mestrado, Mestrado em Informática, UFES, 2005.
- NUNES, V. B., FALBO, R. A., (2006) "Uma Ferramenta de Gerência de Configuração Integrada a um Ambiente de Desenvolvimento de Software". V Simpósio Brasileiro de Qualidade de Software. Vila Velha, Brasil, Maio/Junho 2006. p. 231-247.
- O'LEARY, D. E., STUDER, R., (2001) "Knowledge Management: An Interdisciplinary Approach". IEEE Intelligent Systems, January/February, vol. 16, No. 1, 2001.
- PALACIO, M. O., (2007) "GQA-ODE: Uma Ferramenta de Apoio à Garantia de Qualidade em ODE", Monografia de conclusão do curso de Ciência da Computação, UFES, 2007.
- PFLEEGER, S. L., (1997) Use Realistic, Effective Software Measurement, in: *Constructing Superior Software*, Software Quality Institute Series, The Software Quality Institute.

- PFLEEGER, S. L., (2004) *Engenharia de Software: Teoria e Prática*, 2ª Edição, Prentice Hall.
- POTTER, N. S., SAKRY, M. A. (2002) *Making Process Improvement Work*, 1ª Edição, Addison-Wesley.
- ROCHA, A. R. C., MALDONADO, J. C., WEBER, K. C., (2001) *Qualidade de Software: Teoria e Prática*. São Paulo: Prentice Hall.
- RUY, F. B., (2006) “Semântica em um Ambiente de Desenvolvimento de Software”, Dissertação de Mestrado, Mestrado em Informática, UFES, 2006.
- SALVIANO, C. F., (2006) Uma Proposta Orientada a Perfis de Capacidade de Processo para a Evolução da Melhoria de Processo de Software, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, FEEC/UNICAMP.
- SEI (2006a), CMMI for Development – v1.2, Pittsburgh: Software Engineering Institute.
- SEI (2006b), Standard CMMI Appraisal Method for Process Improvement – v1.2, Pittsburgh: Software Engineering Institute.
- SILVA FILHO, R. C., ROCHA, A. R., TRAVASSOS, G. H., (2007) “Uma Abordagem para Avaliação de Propostas de Melhoria em Processos de Software”, In: Anais do VI Simpósio Brasileiro de Qualidade de Software, SBQS’2007, Porto de Galinhas, Brasil. Anais... 2007, p. 485-500.
- SIQUARA, E. C., (2007) Uma Proposta de Ferramenta para Automação de Avaliações CMMI Baseadas no Método SCAMPI, Dissertação de Mestrado, Universidade Salvador, Salvador.
- SOFTEX (2007a) MPS.BR – Melhoria de Processo do Software Brasileiro: Guia Geral, Versão 1.2.
- SOFTEX (2007b) MPS.BR – Melhoria de Processo do Software Brasileiro: Guia de Avaliação, Versão 1.1.
- SOMMERVILLE, I., RANSOM, J. (2005) “An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement”, ACM Transactions on Software Engineering and Methodology, Volume 14, Issue 1, p. 85 - 117.
- SPYNS, P., MEERSMAN, R., JARRAR, M., (2002) “Data Modelling Versus Ontology Engineering”. SIGMOD Rec., Vol. 31, No. 4. (December 2002), pp. 12-17.
- SQI – SOFTWARE QUALITY INSTITUTE (2007), Appraisal Assistant Beta, Queensland: Griffith University. <<http://www.sqi.gu.edu.au/AppraisalAssistant/>>.
- STAAB, S., STUDER, R., SCHURR, H. P., SURE, Y., (2001) “Knowledge Processes and Ontologies”. IEEE Intelligent Systems, January/February, Vol. 16, No. 1, 2001.
- UMARGI, M., SEAMAN, C., (2005) “Predicting Acceptance of Software Process Improvement”, In: International Conference on Software Engineering, USA, p. 1-6.
- VARKOI, T., (2002) Management of Continuous Software Process Improvement , IEEE, pp. 334-337.
- WANG, Y., KING, G., (2002) Software Engineering Processes: Principles and Applications, CRC Press. (EuroSPI Konferenzbände) 2002.

XAVIER, J. M. C., (2007) “ProEvaluator: Uma Ferramenta para Avaliação de Processos de Software”, Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife.

ZAHARAN, S., (1998) *Software Process Improvement*, Addison-Wesley.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)