UNIVERSIDADE SALVADOR – UNIFACS

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

MESTRADO PROFISSIONAL EM SISTEMAS E COMPUTAÇÃO

THALES CASTELO BRANCO DE CASTRO

UMA ABORDAGEM PARA A INTEGRAÇÃO DE FERRAMENTAS PARA APOIO AOS PROCESSOS DO NÍVEL G DO MPS.BR

Salvador
2009

THALES CASTELO BRANCO DE CASTRO

UMA ABORDAGEM PARA INTEGRAÇÃO DE FERRAMENTAS PARA APOIO AOS PROCESSOS DO NÍVEL G DO MPS.BR

Dissertação apresentada ao Mestrado Profissional em Sistemas e Computação da Universidade Salvador – UNIFACS, como requisito parcial para obtenção do grau de Mestre.

Orientador(a): Profª. Dra. Karina Barreto Villela

Salvador
2009

FICHA CATALOGRÁFICA
(Elaborada pelo Sistema de Bibliotecas da Universidade Salvador - UNIFACS)

Castro, Thales Castelo Branco de
Uma abordagem para a integração de ferramentas para apoio aos processos do nível G do MPS.Br / Thales Castelo Branco de Castro . - 2009.
166 f.

Dissertação (Mestrado) – Universidade Salvador – UNIFACS. Mestrado em Sistemas e Computação, 2009.
Orientador: Profª. Drª. Karina Barreto Villela.

1. Ferramentas. 2. Software integrado. 3. Software - Desenvolvimento. 4. Software - Manutenção. 5. Ontologia. I. Villela, Karina Barreto, orient. II. Título.

CDD: 005.1

TERMO DE APROVAÇÃO

THALES CASTELO BRANCO DE CASTRO

UMA ABORDAGEM PARA INTEGRAÇÃO DE FERRAMENTAS PARA APOIO AOS PROCESSOS DO NÍVEL G DO MPS.BR

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, Universidade Salvador – UNIFACS, pela seguinte banca examinadora:

Profa. Dra. Karina Barreto Villela - Orientadora _____
Doutora em Engenharia de Sistemas e Computação , Universidade Federal do Rio de Janeiro, UFRJ, Brasil.
Universidade Salvador – UNIFACS

Prof. Dr. André Santanchè _____
Doutor em Ciências da Computação, Instituto de Computação, Universidade Estadual de Campinas, UNICAMP, Brasil
Universidade Salvador - UNIFACS

Prof. Dr. Manoel Gomes de Mendonça Neto _____
Doutor em Ciência da Computação, Universidade de Maryland, College Park, Estados Unidos
Universidade Federal da Bahia – UFBA

Profa. Dra. Daniela Barreiro Claro _____
Doutora em Ciência da Computação pela Université d'Angers, França
Universidade Federal da Bahia - UFBA

Salvador, 23 de Abril de 2009

AGRADECIMENTOS

Em primeiro lugar a Deus, por ter me dado o dom da vida e a sede pelo conhecimento.

Um agradecimento especial aos meus pais, Delfino e Eliana, pela educação e grande apoio que eles me ofereceram, sem os quais certamente não teria alcançado a minha formação profissional.

A minha mulher, Juliana, que sempre esteve ao meu lado, me apoiando e enfrentando comigo as dificuldades desses anos de estudo.

A minha bênção de Deus, minha filha Lethicia, uma das grandes motivações para seguir em frente nesta minha caminhada.

A minha família, Caroline, Anderson, Anna Karla, Rodolfo e Cristina, e meus amabilíssimos sobrinhos Paulinho, Mateus, Davi e Arthur, por serem meus grandes companheiros. Esses, sim, são a minha Grande Família.

Aos companheiros do NUPERC e da MTM, onde fiz grandes amizades, que acompanharam comigo esses anos de estudo.

A todos os amigos que fiz durante todos esses anos, e que me ajudaram a fazer de mim a pessoa que sou hoje.

A Karina, pela orientação e pelas várias revisões deste trabalho, não deixando de colocar a mão na massa, quando necessário.

A todos os meus professores, pelos ensinamentos a mim transmitidos.

A todos os meus sinceros agradecimentos.

A vida não dá nem empresta, não se comove nem se apieda. Só atribui aquilo que lhe oferecemos.

Albert Einstein, 1879-1955

RESUMO

Diante de um mercado altamente competitivo, as empresas estão cada vez mais preocupadas em obter certificações/avaliações que comprovem a maturidade de seus processos de software. Ferramentas de software possuem importante papel neste contexto, mas podem encarecer a implantação dos processos devido ao alto custo de aquisição. Principalmente para as empresas de pequeno e médio porte, o caminho mais indicado é a adoção do modelo de referência nacional, o MPS.Br, e a utilização de ferramentas livres, de forma a diminuir os custos envolvidos. No entanto, ferramentas isoladas dificultam a execução dos processos, pois, muitas vezes, é preciso registrar as mesmas informações em ferramentas distintas. O ideal, então, é dispor de ferramentas integradas que apóiem os processos de desenvolvimento e manutenção de software das organizações, dando origem a ambientes de desenvolvimento de software voltados para essas organizações. Para que isso seja possível, é necessário tanto um processo que auxilie a seleção das ferramentas, quanto um modelo que determine as informações a serem compartilhadas e que, ao mesmo tempo, funcione como inter-língua entre as diferentes ferramentas. Diante do exposto, esta dissertação concentrou-se na elaboração de um processo para avaliação e seleção de ferramentas livres, e na elaboração de ontologias que determinem os requisitos de informação para atender o nível G do MPS.Br. A efetiva implementação das ontologias foi considerada parte dos trabalhos futuros. A infra-estrutura fornecida está disponível para apoiar pequenas e médias empresas a atingirem o nível G do MPS.Br e dá um importante passo na direção da integração de ferramentas livres baseada em ontologias.

Palavras-chave: Ferramentas. Software Integrado. Software – Desenvolvimento. Software – Manutenção. Ontologia.

ABSTRACT

Facing a highly competitive market, companies are more and more concerned about obtaining certifications/assessments that provide evidence on the maturity of their software processes. Software tools play a relevant role in this context, but they can also increase the cost of process implementation due to their high acquisition costs. Specifically for small and medium companies, the adoption of the national reference model, MPS.Br, and the use of free tools is recommended in order to diminish costs. However, the usage of stand-alone tools make the process execution difficult, because it is often necessary to entry the same information in several tools. The goal should therefore be to integrate a set of tools that support the software development and maintenance processes in a company, creating a software development environment. In order to do so, it's necessary both a process that supports the selection of such tools and a model that determines the information to be shared by them. This model is expected to be used as an inter-language between different tools. This dissertation aimed at defining a process to guide the assessment and selection of free tools, and constructing ontologies that specify the information requirements for attending the level G of MPS.BR. The effective implementation of the proposed ontologies was considered as future work. The resulting infra-structure supports small and medium companies in reaching level G of MPS.Br and represents an important step towards the integration of free tools based on ontologies.

Key-words: Tools. Integrated Software. Software – Development. Software – Maintenance. . Ontology.

LISTA DE FIGURAS

Figura 1 – Modelo de Qualidade para Qualidade Interna e Externa. 20
Figura 2 – Estrutura da Norma ISO/IEC 12207. 26
Figura 3 – Componentes do Modelo CMMI. Obtido em CMMI 2006. 30
Figura 4 – Tipos de Ontologias. Obtido em GUARINO 1998. 50
Figura 5 – Macro-Atividades do Processo de Seleção 59
Figura 6 – Atividades de Iniciação 60
Figura 7 – Atividades de Identificação 63
Figura 8 – Atividades de Seleção 65
Figura 9 – Escopo do Projeto e Profissionais Envolvidos 90
Figura 10 – Matriz de Rastreabilidade 92
Figura 11 – Classificação, Fontes e Atualização de Requisitos 95
Figura 12 – Avaliação e Comprometimento com Artefatos 98
Figura 13 – Classificação de Critérios e Escalas 100
Figura 14 – Escopo e Processo do Projeto 103
Figura 15 – Estimativas de Projeto 108
Figura 16 – Capacitação Profissional 112
Figura 17 – Envolvimento Profissional 113
Figura 18 – Informações do Projeto 116
Figura 19 – Riscos Associados ao Projeto 118
Figura 20 – Monitoração do Projeto 121

LISTA DE QUADROS

Quadro 1 – Qualidades Internas e Externas presentes na Norma ISO 9126 - 1... 22
Quadro 2– Exemplos de Métricas para Qualidade Interna, Qualidade Externa e Qualidade em Uso... 23
Quadro 3 – Níveis de Maturidade no MPS.Br e respectivos Atributos de Processo. Obtido em MPS.BR 2007 ... 36
Quadro 4 – Metodologias para Elaboração de Ontologias ... 54
Quadro 5 – Requisitos Funcionais para a Gerência de Requisitos ... 69
Quadro 6 – Requisitos Funcionais para a Gerência de Projeto... 71
Quadro 7 – Requisitos Não Funcionais para todas Áreas de Processo ... 73
Quadro 8 – Avaliação das Ferramentas para Gerência de Requisitos segundo os Requisitos Críticos ... 75
Quadro 9 – Avaliação das Ferramentas para Gerência de Requisitos segundo os Requisitos Críticos ... 75
Quadro 10 – Escopo do Projeto e Profissionais Envolvidos: Conceitos ... 91
Quadro 11 – Escopo do Projeto e Profissionais Envolvidos: Relações ... 91
Quadro 12 – Matriz de Rastreabilidade: Conceitos ... 92
Quadro 13 – Matriz de Rastreabilidade: Conceitos ... 93
Quadro 14 – Matriz de Rastreabilidade: Axiomas ... 94
Quadro 15 – Classificação, Fontes e Atualização de Requisitos: Conceitos ... 96
Quadro 16 – Classificação, Fontes e Atualização de Requisitos: Relações ... 96
Quadro 17 – Classificação, Fontes e Atualização de Requisitos: Axiomas... 97
Quadro 18 – Avaliação e Comprometimento com Artefatos: Conceitos... 98
Quadro 19 – Avaliação e Comprometimento com Artefatos: Relações... 98
Quadro 20 – Avaliação e Comprometimento com Artefatos: Axiomas... 99
Quadro 21 – Classificação de Critérios e Escalas: Conceitos... 100
Quadro 22 – Classificação de Critérios e Escalas: Relações... 101
Quadro 23 – Classificação de Critérios e Escalas: Axiomas... 101
Quadro 24 – Escopo e Processo do Projeto: Conceitos ... 104
Quadro 25 – Escopo e Processo do Projeto: Relações ... 105
Quadro 26 – Escopo e Processo do Projeto: Axiomas... 106
Quadro 27 – Estimativas do Projeto: Conceitos ... 109
Quadro 28 – Estimativas do Projeto: Relações ... 110
Quadro 29 – Estimativas do Projeto: Axiomas ... 110
Quadro 30 – Recursos Humanos do Projeto: Conceitos ... 114
Quadro 31 – Recursos Humanos do Projeto: Relações... 114
Quadro 32 – Recursos Humanos do Projeto: Axiomas ... 115
Quadro 33 – Informações do Projeto: Conceitos... 116
Quadro 34 – Informações do Projeto: Relações... 117
Quadro 35 – Informações do Projeto: Axiomas... 117
Quadro 36 – Riscos do Projeto: Conceitos ... 119
Quadro 37 – Riscos do Projeto: Relações ... 119
Quadro 38 – Riscos do Projeto: Axiomas... 119
Quadro 39 – Monitoração do Projeto: Conceitos... 122
Quadro 40 – Monitoração do Projeto: Relações... 122
Quadro 41 – Monitoração do Projeto: Axiomas... 123

LISTA DE TABELAS

Tabela 1 – Avaliação de Requisitos Funcionais para Gerência de Requisitos..... 78
Tabela 2 – Avaliação de Requisitos Funcionais para Gerência de Projetos 79
Tabela 3 – Avaliação de Requisitos Não Funcionais para Gerência de Requisitos 80
Tabela 4 – Avaliação de Requisitos Não Funcionais para Gerência de Projetos.. 81
Tabela 5 – Relatório de Avaliação para Gerência de Requisitos 83
Tabela 6 – Relatório de Avaliação para Gerência de Projeto..... 84

LISTA DE ABREVIATURAS E SIGLAS

CASE	*Computer-Aided Software Engineering*
CMMI	*Capability Maturity Model Integration*
Lingo	Linguagem Gráfica para descrever Ontologias
MPS.Br	Melhoria de Processo de Software Brasileiro
ODE	*Ontology-based software Development Environment*
OML	*Ontology Markup Language*
OntoEdit	*Ontology Editor*
OWL	*Web Ontology Language*
PMI	*Project Management Institute*
UML	*Unified Modeling Language*

SUMÁRIO

CAPÍTULO 1. INTRODUÇÃO 13
1.1 VISÃO GERAL 13
1.2 OBJETIVO DO TRABALHO 15
1.3 RESULTADOS OBTIDOS 15
1.4 ESTRUTURA DA DISSERTAÇÃO 16
CAPÍTULO 2. QUALIDADE DE SOFTWARE 17
2.1 VISÃO GERAL 17
2.2 A NORMA ISO 9126-1 19
2.3 A NORMA ISO 14598 23
2.4 A NORMA ISO 12207 25
2.5 O CMMI 28
2.6 O MPS.BR 33
2.7 GERÊNCIA DE REQUISITOS 37
2.8 GERÊNCIA DE PROJETO 39
2.9 CONCLUSÕES 44
CAPÍTULO 3. ONTOLOGIAS 45
3.1 VISÃO GERAL 45
3.2 ELEMENTOS DE UMA ONTOLOGIA 47
3.3 CLASSIFICAÇÃO DE ONTOLOGIAS 48
3.4 LINGUAGENS PARA A REPRESENTAÇÃO DE ONTOLOGIAS 50
3.5 MÉTODOS PARA A ELABORAÇÃO DE ONTOLOGIAS 51
3.6 CRITÉRIOS DE QUALIDADE PARA A ELABORAÇÃO DE ONTOLOGIAS 55
3.7 ATUAIS APLICAÇÕES NA ENGENHARIA DE SOFTWARE 56
3.8 CONCLUSÕES 57
CAPÍTULO 4. PROCESSO PARA AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS DE SOFTWARE 58
4.1 VISÃO GERAL 58
4.2 DESCRIÇÃO DO PROCESSO 59
4.3 APLICAÇÃO DO PROCESSO 67
4.4 CONCLUSÕES 85
CAPÍTULO 5. ONTOLOGIAS PARA APOIO À GERÊNCIA DE REQUISITOS E DE PROJETO 86
5.1 VISÃO GERAL 86
5.2 PROPÓSITO DAS ONTOLOGIAS 87
5.3 CONSIDERAÇÕES SOBRE CAPTURA E FORMALIZAÇÃO 88
5.4 ONTOLOGIA DE GERÊNCIA DE REQUISITOS 89
5.5 ONTOLOGIA DE GERÊNCIA DE PROJETOS 101
5.6 CONCLUSÕES 123
CAPÍTULO 6. CONCLUSÃO 124
6.1 CONSIDERAÇÕES FINAIS E CONTRIBUIÇÕES 124
6.2 LIMITAÇÕES 126
6.3 TRABALHOS FUTUROS 127

CAPÍTULO 1. INTRODUÇÃO

Neste capítulo são apresentadas as motivações e o objetivo deste trabalho, como também os resultados obtidos e como o mesmo está organizado.

1.1 VISÃO GERAL

Organizações de desenvolvimento de software necessitam gerar produtos com qualidade, de forma a sobreviver em um mercado altamente competitivo. Dispor de processos bem definidos é um passo importante para alcançar tal qualidade. Segundo Rocha, Maldonado e Weber (2001, p.1), “a qualidade dos produtos de software [...] está diretamente relacionada com a qualidade dos processos de software”, o que justifica a grande preocupação das organizações com a melhoria de processos. Algumas organizações optam por obter certificações/avaliações que, além de assegurar a maturidade dos seus processos, tragam vantagem competitiva nos negócios. A mais cobiçada atualmente é a baseada no CMMI (*Capability Maturity Model Integration*), desenvolvido pelo SEI (*Software Engineering Institute*), que tem como objetivo fornecer diretrizes para os processos organizacionais em algumas áreas de conhecimento, dentre elas a Engenharia de Software (SEI, 2002). Na sua representação em estágio, o modelo possui cinco níveis, os quais indicam o grau de maturidade da organização em relação às áreas de conhecimento selecionadas: 1 – Inicial, onde os processos ainda são imprevisíveis e sem controle; 2 – Gerenciado, quando os processos se tornam disciplinados; 3 – Definido, no qual os processos são consistentes e padronizados; 4 – Gerenciado Quantitativamente, quando os processos se tornam previsíveis e controlados; e 5 – Em otimização, que implica em processos aperfeiçoados. Para cada um desses níveis, o modelo fornece orientações e melhores práticas, agrupadas por áreas de processo que devem ser implementadas nas organizações.

No entanto, o CMMI é um modelo internacional, com custo muito alto para empresas de pequeno e médio porte, as quais representam a grande parte das

empresas nacionais que desenvolvem software. Como uma alternativa, foi criado um modelo de referência nacional em 2005, o MPS.Br, que tem ganho força desde então. 72 empresas foram avaliadas até o final de 2007 e mais 53 empresas foram avaliadas no ano de 2008 (MPS.Br, 2007). O MPS.Br está baseado no CMMI, além de nas normas ISO/IEC 12207 (Processo de Ciclo de Vida de Software) e ISO/IEC 15504 (Avaliação de Processo). Ele possui apenas a visão em estágios, começando pelo Nível G (Parcialmente Gerenciado), passando pelos níveis F (Gerenciado), E (Parcialmente Definido), D (Largamente Definido), C (Definido), B (Gerenciado Quantitativamente) e atingindo, finalmente, o nível mais alto, A (Em Otimização). Cada um dos níveis também é composto por áreas de processo, com respectivas orientações e melhores práticas a serem implementadas nas organizações. O MPS.Br é mais fácil de ser adotado e tem custo mais acessível se comparado com o CMMI, pois possui mais estágios e sua estrutura de avaliação é completamente nacional.

Implementar processos em uma organização não é uma tarefa trivial. Cada um dos processos geralmente é complexo, “devido à grande quantidade de atividades a serem executadas e informações envolvidas” (BERTOLLO; FALBO, 2002, p.1). Neste contexto, ferramentas CASE (*Computer-Aided Software Engineering*) são fundamentais, pois, além de facilitarem a execução das atividades, podem auxiliar no entendimento do processo e na institucionalização de suas práticas, oferecendo suporte adequado ao desenvolvimento e à manutenção de sistemas de software. Como tais ferramentas podem encarecer a implantação dos processos, devido ao alto custo de aquisição das mesmas, uma alternativa interessante é a adoção de ferramentas livres. Um estudo feito pelo Serviço Federal de Processamento de Dados (SERPRO) afirma que houve uma redução de custos de cerca de R$ 14,8 milhões a partir da adoção de software livre, o que exigiu investimentos de apenas R$ 396 mil em serviços e treinamento (SOFTWARE LIVRE, 2006). Todavia, restam ainda duas questões: como obter ferramentas livres mais robustas que atendam aos requisitos técnicos e gerenciais; e como prover uma infra-estrutura de integração que viabilize o uso das várias ferramentas necessárias, uma vez que cada uma geralmente se

destina a apoiar apenas um processo ou parte dele. Para tratar a primeira questão, segundo a ISO/IEC (1995), atenção especial deve ser dada à seleção de ferramentas CASE, tanto com relação aos requisitos técnicos quanto aos requisitos gerenciais. A segunda questão é resolvida através de um Ambiente de Desenvolvimento de Software (ADS), o qual se propõe a disponibilizar ferramentas que apóiem todo o processo de desenvolvimento e manutenção (FALBO; MIAN; NATALI, 2001), o que significa que as ferramentas podem possuir um propósito específico, mas devem estar integradas de forma a serem consideradas como um conjunto pelos usuários, fornecendo funcionalidades complementares, sem necessidade de re-trabalho.

1.2 OBJETIVO DO TRABALHO

O objetivo desta dissertação foi dar um passo em direção ao desenvolvimento de ADSs baseados em ferramentas livres. Para tal, decidiu-se definir um processo genérico para avaliação e seleção de ferramentas CASE, e demonstrar a sua utilização na seleção de ferramentas livres para apoiar as práticas do nível G do MPS.Br, que consiste das áreas de processo Gerência de Projeto e Gerência de Requisitos. Além disso, para definir os requisitos de informação referentes ao nível G do MPS.Br e, ao mesmo tempo, possibilitar a futura integração de ferramentas, decidiu-se elaborar ontologias que modelassem os conceitos relevantes às áreas de processo desse nível.

1.3 RESULTADOS OBTIDOS

O processo genérico para avaliação e seleção de ferramentas CASE foi executado para permitir a avaliação e seleção de ferramentas livres para apoio à Gerência de Projeto e Gerência de Requisitos, o que significa que um conjunto de requisitos funcionais e não funcionais referentes a estas áreas de processo foi identificado e priorizado. Este conjunto de requisitos está disponível para ser reutilizado em contexto similares. Além disso, 34 ferramentas para Gerência de

Projeto e 33 ferramentas para Gerência de Requisitos foram consideradas no processo, das quais 6 e 3 ferramentas das respectivas áreas de processo foram pré-selecionadas segundo os requisitos estabelecidos. O processo resultou na seleção final da melhor ferramenta para apoiar cada área de processo. As tabelas fornecidas para comparação das ferramentas podem auxiliar empresas interessadas na seleção de ferramentas livres para essas áreas de processo. As ontologias de Gerência de Projeto e de Gerência de Requisitos, também resultados desta dissertação, fornecem juntas 94 conceitos, 77 relações e 65 axiomas que definem o vocabulário a ser compartilhado entre as ferramentas.

1.4 ESTRUTURA DA DISSERTAÇÃO

O restante desta dissertação está organizada da seguinte forma: o capítulo 2 fornece uma visão geral sobre Qualidade de Software e os seus principais modelos e normas; o capítulo 3 tem como foco o tema ontologias; o capítulo 4 apresenta o processo definido para avaliação e seleção de ferramentas CASE, e o resultado de sua execução visando a seleção de ferramentas livres para apoiar a Gerência de Projeto e a Gerência de Requisitos; o capítulo 5 descreve as ontologias de Gerência de Projeto e de Gerência de Requisitos que foram elaboradas para contemplar o nível G do MPS.Br; as considerações finais e as contribuições desta dissertação, assim como suas limitações e trabalhos futuros, são apresentados no capítulo 6.

CAPÍTULO 2. QUALIDADE DE SOFTWARE

Este capítulo fornece uma visão geral sobre qualidade de software e apresenta o conjunto de normas e modelos relevantes para este trabalho: a ISO/IEC 9126, a ISO/IEC 14598, a ISO/IEC 12207, o CMMI e o MPS.Br.

2.1 VISÃO GERAL

Estudos feitos afirmam que grande parte da população mundial depende da execução de sistemas para a realização de tarefas cotidianas. “Se alguns sistemas de uso global deixarem de funcionar, aproximadamente 40% da população mundial sofrerá as conseqüências do problema.” (ROCHA; MALDONADO; WEBER, 2001, p. 1). Neste contexto, muitos dos sistemas de software são críticos para o sucesso de uma organização ou para a segurança humana, sendo de fundamental importância que executem com um nível de desempenho adequado. Este nível de desempenho depende diretamente da qualidade do sistema que foi elaborado, ou seja, “do conjunto de características que devem estar presentes em um determinado grau, de modo que o software atenda às necessidades de seus usuários.” (VILLELA, 2006, p. 1).

Um aspecto a destacar na declaração acima é que geralmente as características não estão todas presentes em grau máximo em um software. “Em um software de tempo real, por exemplo, eficiência é muito importante e, para melhorar a eficiência, podem ser adotados uma linguagem de programação e um algoritmo que prejudicarão a manutenibilidade.” (VILLELA, 2006, p. 2). Desta forma, a finalidade de um software com qualidade é alcançar a características necessárias e suficientes que possam satisfazer às reais necessidades dos usuários, quando o produto for utilizado no contexto de uso especificado (ABNT, 2002a).

A questão torna-se, então, como desenvolver o software de modo que o mesmo possua as características de qualidade desejadas. A ISO/IEC 9126

(*Engenharia de Software – Qualidade do Produto*) provê um conjunto de características e métricas que pode ser utilizado para elaborar um modelo de qualidade, contendo os requisitos de qualidade do software e as métricas a serem utilizadas para controlar e verificar que os mesmos estão sendo atendidos. Para complementar, a norma ISO/IEC 14598 (*Tecnologia de Informação – Avaliação de Produtos de Software*) fornece um guia para avaliação da qualidade de produtos de software sob diferentes pontos de vista: desenvolvedor, adquirente e certificador (ISO/IEC, 1998).

Por outro lado, é frequentemente mencionado que a qualidade de um software depende diretamente da qualidade dos processos utilizados no seu desenvolvimento (ROCHA; MALDONADO; WEBER, 2001). Por ser uma abordagem com efeito simultâneo em diferentes projetos, grande atenção tem sido dada à qualidade dos processos de sofware.

De acordo com Fuggetta (2000, p. 3), um processo “é um conjunto de passos ordenados intencionados a atingir um objetivo; em engenharia de software, o objetivo é construir um produto de software ou melhorar um existente”. Processos de software devem estar devidamente caracterizados, qualquer que seja o tipo de projeto, sua área de atuação, seu tamanho ou complexidade. Em um processo de desenvolvimento, existem basicamente três fases que devem ser consideradas: *Definição, Desenvolvimento* e *Manutenção* (ROCHA; MALDONADO; WEBER, 2001), mas a existência dessas fases, individualmente, não garantem a execução de um processo com qualidade. Deve-se levar em consideração também fatores não técnicos que influem diretamente na qualidade dos processos, como as características da organização, o domínio da aplicação e a escolha do pessoal técnico (ROCHA, MALDONADO; WEBER, 2001). A norma ISO/IEC 12207 (*Tecnologia de Informação – Processos de Ciclo de Vida do Software*) contém um conjunto de processos que ajudam a definir, e apoiar a execução e a finalização de processos diretamente relacionados a um projeto de software; como também de processos voltados para a organização. A norma também propõe atividades para implementação de cada um desses processos.

Além disso, existem dois modelos relevantes para esta dissertação, os quais prescrevem um conjunto de melhores práticas/resultados a serem considerados quando um processo estiver sendo definido e implementado: o CMMI – (*Capability Maturity Model* – Modelo de Capacidade de Maturidade) e o MPS.Br (*Melhoria de Processo de Software Brasileiro*).

Existem outras normas relacionadas a qualidade de produto e de processo que não são discutidas neste capítulo, mas são mencionadas a seguir. Com relação a qualidade de produto, tem-se ainda a ISO/IEC 12119 (*Tecnologia de Informação – Pacotes de Software – Testes e Requisitos de Qualidade*), que estabelece requisitos e instruções de como testar um pacote de software de acordo com os requisitos; e a ISO/IEC 25000 *(Software Engineering -- Software Product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE)*, que foi criada em 2005 para organizar os requisitos de qualidade das normas ISO 9126 e ISO 14598 em uma única série de normas com o objetivo de "apoiar aos que desenvolvem ou adquirem produtos com a especificação e avaliação de requisitos de qualidade." (ISO/IEC, 2005, p. v). Relacionadas a qualidade de processo, existem as normas ISO 15504 (*Tecnologia de Informação – Avaliação de Processo*), que contém as atividades relacionadas à avaliação da implementação, em uma organização, dos processos propostos na ISO/IEC 12207; e a ISO 9001 (*Sistemas de Gerenciamento de Qualidade*), também utilizada por organizações que desejem aferir um grau de qualidade a seus produtos com base na qualidade de seus processos, sem considerar uma atividade de negócio ou área de conhecimento específicos. A ISO 90003 *(Engenharia de Software – Guia para Aplicação da Norma ISO 9001:2000 para Software)* é a norma que oferece uma guia para aplicação da ISO 9001 na aquisição, desenvolvimento, fornecimento e operação de software e serviços relacionados (ISO/IEC, 2004).

2.2 A NORMA ISO 9126-1

A norma ISO/IEC 9126 (*Engenharia de Software – Qualidade do Produto*) possui quatro partes: ISO/IEC 9126-1: Modelo de Qualidade; ISO/IEC 9126-2:

Métricas Externas; ISO/IEC 9126-3: Métricas Internas; e ISO/IEC 9126-4: Métricas de Qualidade em Uso. A primeira parte da norma (ISO 9126-1), por sua vez, encontra-se dividida em duas seções: *Qualidade Interna e Externa* e *Qualidade em Uso*. Qualidade Interna refere-se as características do produto de software do ponto de vista interno (Ex: modularidade), podendo ser avaliada em produtos intermediários, tais como código-fonte, requisitos, documentação etc; Qualidade Externa refere-se as características do produto de software do ponto de vista externo (Ex: desempenho, manutenibilidade), podendo ser avaliada apenas após o produto de software ser concluído; e Qualidade em Uso refere-se a qualidade do produto de software do ponto de vista do usuário (Ex: eficiência no uso, produtividade), ou seja, indica o grau em que os usuários conseguem atingir seus objetivos em um determinado ambiente e contexto de uso (ABNT, 2002a).

Na seção *Qualidade Interna e Externa* são definidas seis características de qualidade, que podem ser tanto internas quanto externas, e as quais são subdivididas em sub-características (Figura 1).

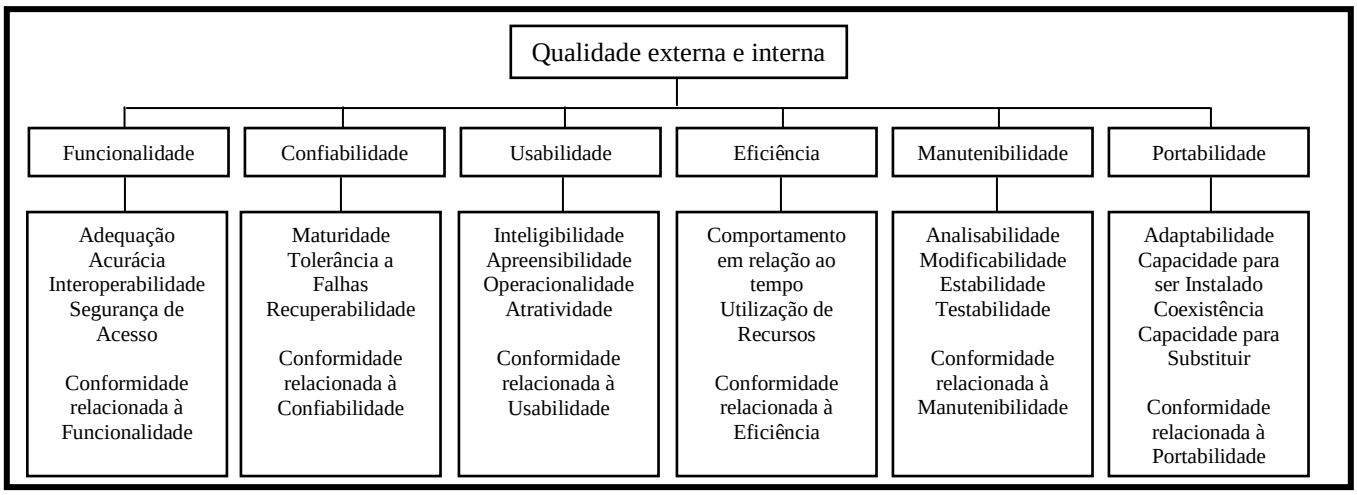


Figura 1 – Modelo de Qualidade para Qualidade Interna e Externa
Nota: Adaptado de ABNT (2002a).

No Quadro 1 são apresentadas as definições das características e sub-características acima.

***Funcionalidade*:** capacidade de prover funções que atendam às necessidades explícitas ou implícitas, quando o software estiver sendo utilizado sob as condições especificadas	
Adequação	Capacidade de de prover um conjunto apropriado de funções para tarefas e objetivos dos usuários especificados
Acurácia	Capacidade de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados
Interoperabilidade	Capacidade de interagir com um ou mais sistemas especificados
Segurança de Acesso	Capacidade de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los, e que não seja negado acesso às pessoas ou sistemas autorizados
Conformidade relacionada à Funcionalidade	Capacidade de estar de acordo com as normas, convenções ou regulamentações previstas em lei e prescrições similares relacionadas à funcionalidade
***Confiabilidade*:** capacidade de manter um nível de desempenho especificado, quando usado nas condições especificadas	
Maturidade	Capacidade de evitar falhas decorrentes de defeitos no software
Tolerância a Falhas	Capacidade de manter um nível de desempenho especificado em casos de defeitos no software ou violação da interface especificada
Recuperabilidade	Capacidade de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha
Conformidade relacionada à Confiabilidade	Capacidade de estar de acordo com as normas, convenções ou regulamentações relacionadas à confiabilidade
***Usabilidade*:** capacidade de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob as condições especificadas	
Inteligibilidade	Capacidade de possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas
Apreensibilidade	Capacidade de possibilitar ao usuário aprender a sua aplicação
Operacionalidade	Capacidade de possibilitar ao usuário operá-lo e controlá-lo
Atratividade	Capacidade de ser atraente ao usuário
Conformidade relacionada à Usabilidade	Capacidade de estar de acordo com as normas, convenções, guias de estilo ou regulamentações relacionadas à usabilidade
***Eficiência*:** capacidade de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob as condições especificadas	
Comportamento em relação ao Tempo	Capacidade de fornecer tempos de resposta e de processamento, além de taxas de transferência apropriados, quando o software executa suas funções, sob condições estabelecidas
Utilização de Recursos	Capacidade de usar tipo e quantidades apropriados de recursos, quando o software executa suas funções sob condições estabelecidas
Conformidade relacionada à Eficiência	Capacidade de estar de acordo com as normas e convenções relacionadas à eficiência

***Manutenibilidade*:** capacidade de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software	
Analisabilidade	Capacidade de permitir o diagnóstico de defeitos ou causas de falhas no software, ou a identificação de partes a serem modificadas
Modificabilidade	Capacidade de permitir que uma modificação especificada seja implementada
Estabilidade	Capacidade de evitar efeitos inesperados decorrentes de modificações no software
Testabilidade	Capacidade de permitir que o software, quando modificado, seja validado
Conformidade relacionada à Manutenibilidade	Capacidade de estar de acordo com as normas e convenções relacionadas à manutenibilidade
***Portabilidade*:** capacidade do produto de software de ser transferido de um ambiente para outro	
Adaptabilidade	Capacidade de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado
Capacidade para ser instalado	Capacidade de ser instalado em um ambiente especificado
Coexistência	Capacidade de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos comuns
Capacidade para substituir	Capacidade de ser usado em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente
Conformidade relacionada à Portabilidade	Capacidade de estar de acordo com as normas e convenções relacionadas à portabilidade

Quadro 1 – Qualidades Internas e Externas presentes na Norma ISO 9126 - 1

As características de qualidade em uso, definidas na seção *Qualidade em Uso* da ISO/IEC 9126-1 são 4:

- *Eficácia*: capacidade de permitir que usuários atinjam metas especificadas com acurácia e completitude;
- *Produtividade*: capacidade de permitir que seus usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida;
- *Segurança*: capacidade de apresentar níveis aceitáveis de riscos de danos a pessoas, negócios, software, propriedades ou ao ambiente;
- *Satisfação*: capacidade de satisfazer usuários.

Para exemplificar, algumas métricas propostas nas partes ISO/IEC 9126-2, ISO/IEC 9126-3 e ISO/IEC 9126-4, que tratam, respectivamente, de métricas externas, internas e de qualidade de uso, são apresentadas no Quadro 2.

Métricas Externas			
Sub-característica:	Adequação	*Métrica*:	Adequação Funcional
Descrição:	Quão adequadas estão as funções avaliadas?		
Fórmula:	X=1-A/B , onde A= Número de funções nos quais problemas foram detectados, e B= Número de funções avaliadas		
Interpretação:	0 <= X <= 1; quanto mais perto de 1, mais adequado		
Sub-característica:	Acurácia	*Métrica*:	Expectativa de Acurácia
Descrição:	As diferenças entre os resultados esperados e os avaliados são aceitáveis?		
Fórmula:	X=A/B , onde A = Número de casos encontrados pelo usuário com diferença entre o resultado esperado e o avaliado, e T = Tempo de Operação		
Interpretação:	0 <= X; quanto mais perto de 0, melhor		
Métricas Internas			
Sub-característica:	Adequação	*Métrica*:	Completude
Descrição:	Quão completa é a implementação funcional?		
Fórmula:	X=1 - A/B , onde A = Número de funções não presentes detectadas na avaliação, e B = Número de funções descritas na especificação de requisitos		
Interpretação:	0 <= X <= 1; quanto mais perto de 1, mais completa		
Sub-característica:	Acurácia	*Métrica*:	Precisão
Descrição:	Quão completa foi a implementação de níveis específicos de precisão para os itens de dados?		
Fórmula:	X=A/B , onde A = Número de itens de dados implementados com níveis específicos de precisão, confirmados na avaliação, e B = Número de itens de dados que requerem níveis específicos de precisão		
Interpretação:	0 <= X <= 1; quanto mais perto de 1, mais completo		
Métricas de Qualidade em Uso			
Sub-característica:	Eficiência	*Métrica*:	Freqüência de Erros
Descrição:	Qual é a freqüência de erros?		
Fórmula:	X=A/T , onde A = Número de erros cometidos pelo usuário, e T = Tempo ou Número de Tarefas		
Interpretação:	0 <= X; quanto mais perto de 0, melhor		
Sub-característica:	Produtividade	*Métrica*:	Tempo da Tarefa
Descrição:	Qual é a eficiência dos usuários?		
Fórmula:	X=M1/T , onde M1 = Efetividade da Tarefa, e T = Tempo da Tarefa		
Interpretação:	0 <= X; quanto maior X, melhor		

Quadro 2– Exemplos de Métricas para Qualidade Interna, Qualidade Externa e Qualidade em Uso

2.3 A NORMA ISO 14598

A norma ISO/IEC 14598 (*Tecnologia de Informação – Avaliação de Produtos de Software*) encontra-se dividida em 6 partes: a 14598-1, Visão Geral, que ensina a utilizar as outras normas do grupo; a 14598-2, Planejamento e Gerenciamento, que mostra de forma geral como realizar uma avaliação; a 14598-3 até a 14598-5, respectivamente Guia para Desenvolvedores, Guia para Aquisição e Guia para Avaliação, que mostram como avaliar o produto de software do ponto de vista de quem desenvolve, quer adquirir ou precisa certificar o produto de software; e a

14598-6, Módulos de Avaliação, que fornece detalhes sobre como avaliar cada característica (ISO/IEC, 1998). Nesta seção, devido a sua relevância para esta dissertação, a parte 14598-5: Guia de Avaliação é descrita resumidamente.

O processo proposto pela ISO 14598-5 é abstrato e aplicável dentro do ciclo de vida de processos definidos na ISO/IEC 12207. A norma pode também ser usada para testar conformidades a padrões, como a ISO/IEC 12119 (*Tecnologia de Informação – Pacotes de Software – Testes e Requisitos de Qualidade*). As atividades propostas no processo são:

1. **Estabelecer os Requisitos da Avaliação**: onde são descritos os objetivos da avaliação, de acordo com o uso pretendido do produto e os riscos associados. Possui como sub-atividades: analisar as razões para a avaliação, descrever os requisitos, definir a cobertura da avaliação, explicar seu rigor e a confiabilidade, e obter acordo sobre os requisitos da avaliação;
2. **Especificar a Avaliação**: inclui a definição do escopo da avaliação e das medidas a serem utilizadas na avaliação dos produtos e seus componentes. Possui como sub-atividades: analisar a descrição do produto, especificar as medições, verificar a especificação de acordo com os requisitos da avaliação, e obter acordo sobre a especificação;
3. **Planejar a Avaliação**: refere-se a documentação dos procedimentos a serem utilizados pelo avaliador para obtenção das medidas, descrevendo os recursos necessários e alocando-os às diversas ações. Possui como sub-atividades: analisar as restrições técnicas, documentar os procedimentos da avaliação, elaborar o plano de avaliação, estabelecer o cronograma para avaliações, e obter acordo sobre o plano de avaliação;
4. **Executar a Avaliação**: refere-se à obtenção dos resultados da aplicação das medidas aos produtos e componentes, e geração dos relatórios. Possui como sub-atividades: gerenciar componentes do produto e documentos relacionados, gerenciar dados da avaliação, gerenciar uso das ferramentas, controlar avaliação no ambiente do

desenvolvedor ou da operação, revisar resultados intermediários e finais, e elaborar o relatório da avaliação; e

5. **Conclusão da Avaliação**: consiste na revisão dos relatórios e sua posterior publicação. Possui como sub-atividades: revisar em conjunto com o solicitante o relatório da avaliação; devolver documentos submetidos, arquivar sob período especificado ou destrui-los; arquivar o relatório da avaliação e os registros da avaliação por períodos especificados; e arquivar por período especificado ou destruir demais dados.

2.4 A NORMA ISO 12207

A norma ISO/IEC 12207 (*Tecnologia da Informação – Processos de ciclo de vida de software*) visa prover uma estrutura comum para os processos de ciclo de vida do software, desde o levantamento de idéias até a descontinuação do produto, e os divide em processos fundamentais, de apoio e organizacionais, conforme mostrado na Figura 2 (ABNT, 1998). Cada processo é definido em termos de atividades e tarefas.

Os processos fundamentais são aqueles que atendem às partes fundamentais do ciclo de vida do software, ou seja, as partes responsáveis por iniciar ou executar o desenvolvimento, a operação e/ou a manutenção do software (ABNT, 1998). Tais processos são listados a seguir.

1. Aquisição, cujas as atividades são: *Iniciação*, *Elaboração do Pedido de Proposta*, *Monitoração do Fornecedor*, e *Aceitação e Conclusão*;

2. Fornecimento, que inclui as atividades de *Iniciação*, *Preparação da Resposta*, *Contrato*, *Planejamento*, *Execução e Controle*, *Revisão e Avaliação*, e *Entrega e Conclusão*;

3. Desenvolvimento, composto das atividades *Análise de Requisitos do Sistema*; *Projeto de Arquitetura do Sistema, Análise dos Requisitos do Software*, *Projeto da Arquitetura*, *Projeto Detalhado*, *Codificação e Testes*, *Integração do Software*, *Teste de Qualificação do Software*, *Integração do*

Sistema, *Teste de Qualificação do Sistema*, *Instalação*, e *Apoio à Aceitação do Software*;

4. **Operação**, cujas atividades são *Teste do Sistema*, *Operação do Produto* e *Suporte Operacional*;

5. **Manutenção**, composto das atividades *Análise do Problema e da Modificação*, *Implementação da Modificação*, *Revisão e Aceitação da Modificação*, *Migração* e *Descontinuação do Software*.

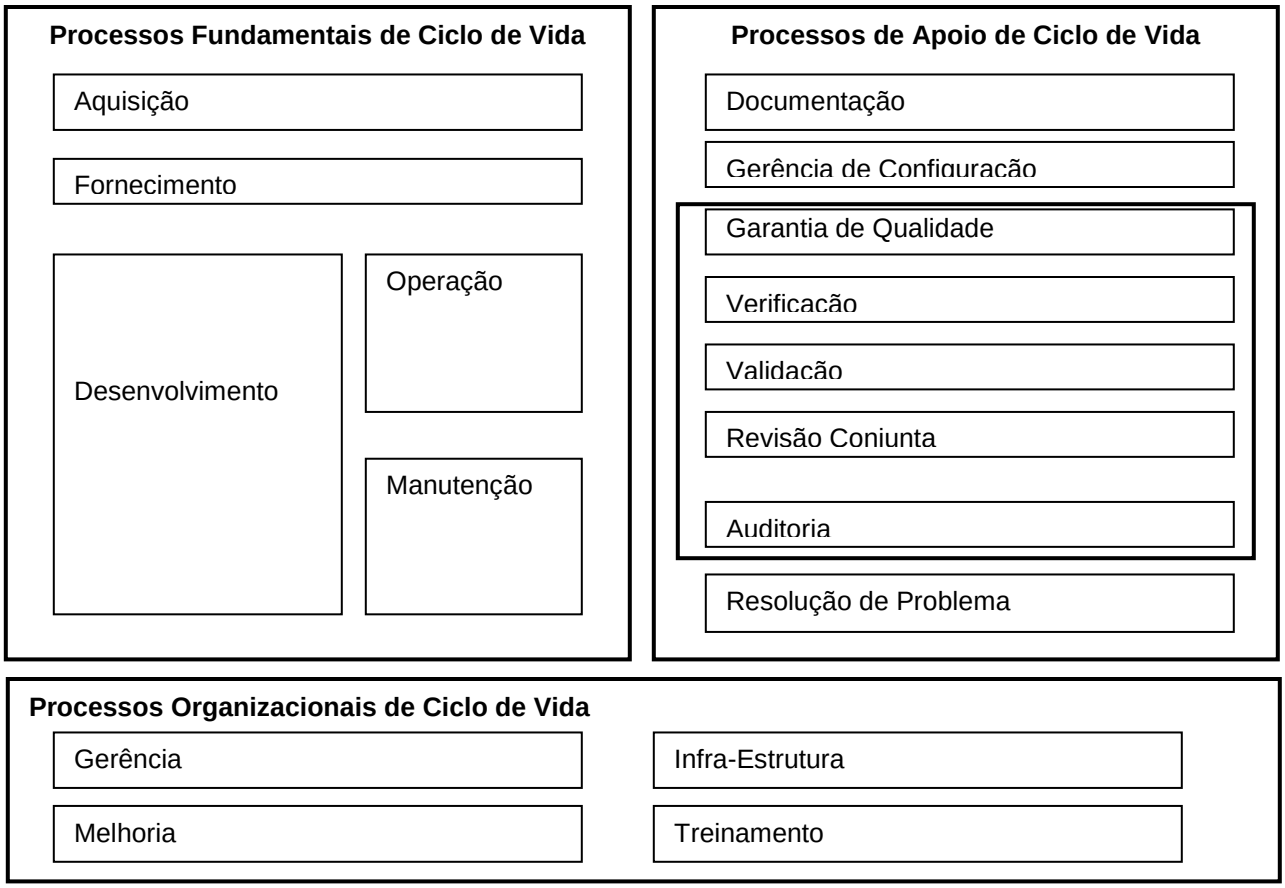


Figura 2 – Estrutura da Norma ISO/IEC 12207
Nota: Adaptado de ABNT (1998).

O segundo grupo de processos, os processos de apoio, são responsáveis por auxiliar os demais processos, como parte integrante, contribuindo para seu sucesso e qualidade (ABNT, 1998), sendo executados apenas quando necessários. Os processos de apoio propostos na (ABNT, 1998) são:

1. **Documentação**, que inclui as atividades de *Projeto e Desenvolvimento*, *Produção* e *Manutenção* da documentação;

2. **Gerência de Configuração**, composto das atividades *Identificação da Configuração, Controle da Configuração*, *Relato de Situação da Configuração*, *Avaliação da Configuração* e *Liberação e Distribuição*;
3. **Garantia de Qualidade**, o qual pode ser tanto um processo interno quanto externo à organização e tem como atividades: *Garantia do Produto*, *Garantia do Processo* e *Sistemas de Garantia de Qualidade*;
4. **Verificação**, que não se encontra decomposto em termos de atividades, mas diretamente descrito em termos de tarefas relacionadas à verificação de contrato, do processo, dos requisitos, do projeto, código, integração e documentação;
5. **Validação**, de forma similar ao processo de verificação, encontra-se diretamente descrito em termos de tarefas como a preparação dos casos de teste, sua condução e a validação para o uso pretendido;
6. **Revisão Conjunta**, o qual inclui atividades de *Revisões de Gerenciamento de Projeto* e *Revisões Técnicas*;
7. **Auditoria**, assim como os processos de verificação e validação, não possui sub-atividades, mas contém tarefas que buscam assegurar que os produtos codificados reflitam a documentação do projeto, que a documentação do usuário esteja aderente a padrões, que os produtos sejam testados com sucesso, dentre várias outras tarefas;
8. **Resolução de Problema**, que também não possui sub-atividades, mas as tarefas devem ser feitas de modo que seja elaborado um relatório com o(s) problema(s) detectado(s), contendo análise, causa e como será resolvido.

O terceiro e último bloco de processos corresponde aos organizacionais de ciclo de vida, que são empregados pela organização com o objetivo de fornecer os processos e o pessoal associado, bem como garantir a melhoria desses processos. “Eles são tipicamente empregados fora do domínio de projetos e contratos específicos; entretanto, ensinamentos destes projetos e contratos contribuem para a melhoria da organização.” (ABNT, 1998, p.6). São quatro os processos organizacionais:

1. **Gerência**, cujas atividades propostas são: *Iniciação e Definição do Escopo*, *Planejamento*, *Execução e Controle*, *Revisão e Avaliação* e *Conclusão*;
2. **Melhoria**, que inclui as atividades de *Estabelecimento*, *Avaliação* e *Melhoria do Processo*;
3. **Infra-Estrutura**, composto das atividades de *Estabelecimento da Infra-Estrutura* e sua *Manutenção*;
4. **Treinamento**, cuja algumas das atividades são: *Desenvolvimento do Material de Treinamento* e *Implementação do Plano de Treinamento*.

A ISO/IEC 12207, no entanto, não cobre metodologias, métodos ou métricas que devem ser abordados pelos processos, bem como os procedimentos, técnicas, ferramentas e ambientes que podem ser utilizadas nas organizações. Sua principal função é apenas estabelecer uma estrutura comum para que todos os interessados (adquirente, fornecedor, desenvolvedor, mantenedor, operador, gerentes e técnicos) envolvidos com o desenvolvimento de software tenham uma linguagem comum do processo empregado (ABNT, 1998).

2.5 O CMMI

O modelo CMMI foi desenvolvido pelo SEI (*Software Engineering Institute* – Instituto de Engenharia de Software) e contém um conjunto de melhores práticas para processos da área de engenharia de software, podendo ser utilizado para guiar as organizações na melhoria de seus processos.

Inicialmente, foram desenvolvidos CMM's (*Capability Maturity Model* – Modelos de Capacidade de Maturidade) para várias disciplinas, dentre elas, Engenharia de Sistemas, Engenharia de Software, Aquisição de Software, Desenvolvimento e Gerência da Força de Trabalho e Desenvolvimento Integrado de Produtos e Processos. No entanto, quando a organização decidia concentrar seus esforços em duas ou mais disciplinas, a utilização de diferentes modelos era

problemática, pois os modelos possuiam arquitetura, conteúdo e abordagem diferentes. Além disso, era mais custoso para a organização ministrar treinamentos e avaliações em cada modelo específico. O CMMI (*Capability Maturity Model Integration* – Integração dos Modelos de Capacidade de Maturidade) surgiu para resolver o problema ao combinar os modelos básicos em um único framework (SEI, 2002).

O CMMI é organizado em áreas de processo, onde uma área de processo "é um grupo de práticas relacionadas em uma área que, quando executadas de forma coletiva, satisfazem um conjunto de metas consideradas importantes para trazer uma melhoria significativa naquela área" (SEI, 2006, p. iv). Cada área de processo possui objetivos genéricos e específicos. Os objetivos genéricos, assim chamados por se aplicarem a várias áreas de processo, abrangem as práticas genéricas a serem atendidas pela organização. Já os específicos se aplicam apenas a uma área de processo e contém características que devem ser implementadas para satisfazer aquela área de processo (SEI, 2006). A Figura 3 mostra os componentes do modelo.

As áreas de processo do CMMI podem ser abordadas através de duas representações: a representação contínua e a representação em estágio. Na representação contínua, a organização escolhe uma (ou mais) área(s) de processo em que deseja realizar uma melhoria, se especializando nessas áreas de processo continuamente. Por exemplo, uma organização mais interessada no gerenciamento de seus riscos e na integração de componentes de produtos pode escolher a representação contínua e se dedicar a atender os objetivos das áreas de processo Gerência de Riscos e Integração de Produtos. Uma das vantagens da utilização da representação contínua é permitir que a organização selecione uma seqüência de melhorias que melhor atende aos seus objetivos de negócio (SEI, 2006). Na representação em estágio, a organização deseja melhorar a capacidade de desenvolvimento de produtos da companhia como um todo, seguindo um caminho já percorrido por muitas organizações. Cada nível representa o nível de maturidade da organização no desenvolvimento de produtos e engloba um conjunto de áreas de processo.

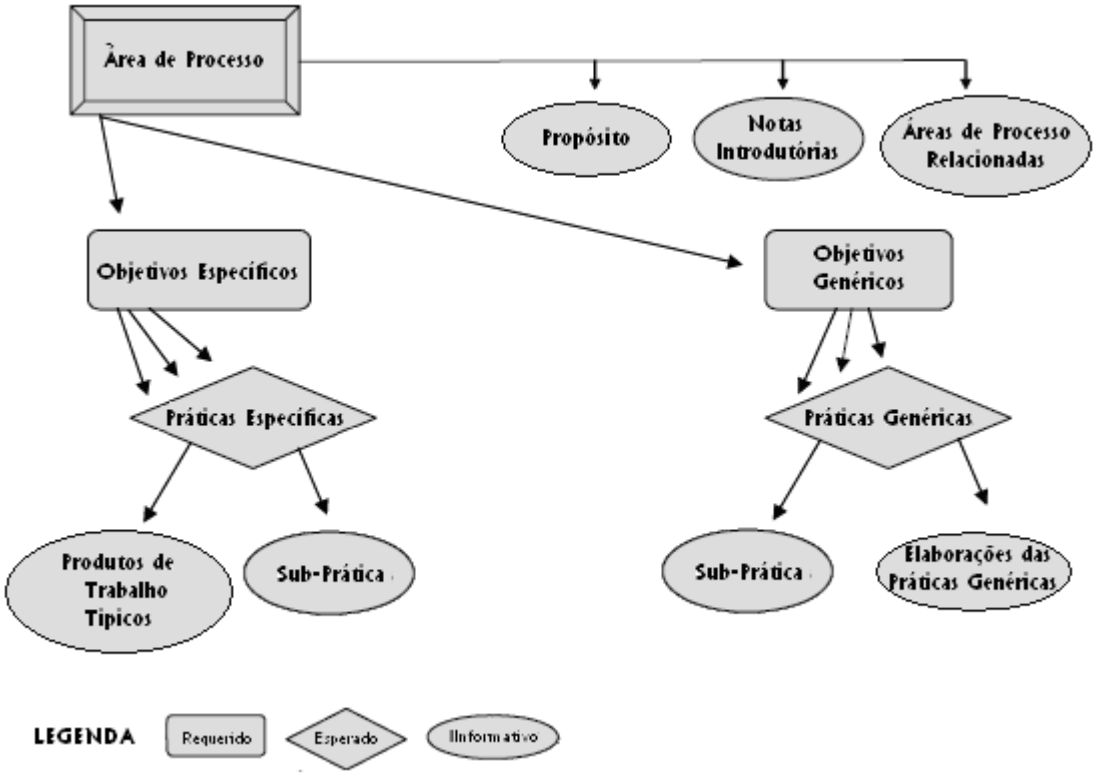


Figura 3 – Componentes do Modelo CMMI
Nota: Adaptado de SEI (2006).

No **Nível 1**, **Inicial**, os processos ainda são imprevisíveis e sem controle. Não existe um ambiente estável de desenvolvimento e o sucesso dessas organizações depende diretamente da competência das pessoas envolvidas nos projetos. Como é o nível inicial, no qual todas as empresas sem avaliação CMMI estão, não existem áreas de processo a serem implementadas.

No **Nível 2**, **Gerenciado,** os processos se tornam disciplinados, ou seja, são planejados, executados, medidos e controlados. Neste nível, existem sete áreas de processo que devem ser implementadas, as quais são:

- **Gerência de Requisitos:** na qual o objetivo é gerenciar os componentes de produtos e componentes de produtos, bem como identificar inconsistências entre os requisitos e os planos e produtos de trabalho (SEI, 2006);

- **Planejamento de Projeto:** a proposta dessa área de processo é estabelecer e manter planos que definam as atividades do projeto;
- **Controle e Monitoramento de Projeto:** tem como objetivo possibilitar o entendimento do progresso do projeto, realizando ações corretivas se necessário;
- **Gerência de Acordo com Fornecedor:** na qual a proposta é gerenciar a aquisição de produtos de fornecedores;
- **Medição e Análise:** tem como objetivo desenvolver e sustentar uma capacidade de medição para dar apoio a necessidades de medição;
- **Garantia de Qualidade de Processos e Produtos:** a proposta é prover a equipe e a gerência com visão objetiva dos processos e produtos de trabalho;
- **Gerência de Configuração:** na qual o objetivo é estabelecer e manter a integridade da configuração dos produtos.

O **Nível 3**, **Definido**, é atingido quando a organização já possui os processos consistentes e descritos em padrões, procedimentos, métodos e ferramentas. Este nível adiciona 13 áreas de processo:

- **Desenvolvimento de Requisitos**: tem como objetivo produzir e analisar requisitos de clientes, produtos e componentes de produtos;
- **Soluções Técnicas**: na qual o objetivo é modelar, desenvolver e implementar soluções para os requisitos;
- **Integração de Produto**: cuja a proposta é planejar a integração dos componentes dos produtos, assegurando que este funciona adequadamente quando os componentes são integrados, visando posterior distribuição do produto;
- **Verificação**: tem como objetivo verificar que o produto foi construído de forma correta;

- **Validação**: tem como objetivo verificar se o produto correto foi construído;
- **Foco do Processo Organizacional**: cujo objetivo é planejar e implementar melhorias nos processos organizacionais, baseado em seus pontos fortes e fracos;
- **Definição do Processo Organizacional**: na qual o objetivo é estabelecer um conjunto de processos que possam ser usados pela organização;
- **Treinamento Organizacional**: cuja proposta é desenvolver conhecimentos e habilidades nas pessoas para que elas possam executar seus papéis de forma eficaz e eficiente;
- **Gerência de Projeto Integrado**: tem como objetivo estabelecer e gerenciar os projetos e o envolvimento dos *stakeholders* de acordo com o processo definido e que deve ser gerenciado de forma integrada;
- **Gerência de Riscos**: na qual o objetivo é identificar potenciais problemas antes que eles ocorram;
- **Análise de Decisões e Resoluções**: tem como objetivo analisar possíveis decisões através de processos de avaliação formais que identificam as alternativas e utilizam critérios.

No **Nível 4, Gerenciado Quantitativamente,** os processos se tornam previsíveis e controlados. Este nível requer mais duas áreas de processo:

- **Desempenho do Processo Organizacional**: cujo objetivo é estabelecer e manter um entendimento quantitativo do desempenho dos processos da organização, disponibilizando dados, baselines e modelos de performance;
- **Gerência Quantitativa de Projeto**: tem como objetivo gerenciar quantitativamente os processos definidos para o projeto para assegurar a qualidade estabelecida para o projeto e os objetivos de desempenho do processo.

O **Nível 5, Em otimização,** implica em processos aperfeiçoados e adiciona mais duas áreas de processo:

- **Inovação e Desenvolvimento Organizacional**: sua proposta é selecionar e implantar melhorias incrementais e inovadoras que melhorem os processos e as tecnologias organizacionais;
- **Análise e Resolução de Causas**: tem como objetivo identificar causas de defeitos ou outros problemas e tomar ações corretivas para prevenir ocorrências futuras destes.

Quando uma organização atinge um determinado nível, ela deve necessariamente cobrir, além dos requisitos das áreas de processo do nível desejado, os requisitos das áreas de processo do nível anterior.

2.6 O MPS.BR

O MPS.Br é um programa para Melhoria de Processo de Software Brasileiro, mas que já está sendo implantado em países com Chile, Argentina e México. O modelo de referência proposto foi publicado em 2005 pela Softex (Associação para Promoção da Excelência do Software Brasileiro), a qual é a coordenadora do programa, e foi desenvolvido em conjunto com universidades, grupos de pesquisa e empresas de desenvolvimento (MPS.Br, 2007).

O MPS.Br utiliza como base técnica os requisitos de processo definidos por normas e modelos internacionais de definição e avaliação de processos de software (a NBR ISO/IEC 12207 e as emendas 1 e 2 da norma internacional ISO/IEC 12207, a ISO/IEC 15504 e o CMMI), e tem como objetivo ser adequado para empresas com diferentes tamanhos e características. No entanto, propõe uma estruturação do conteúdo especialmente adequada ao contexto de micro, pequenas e médias empresas (MPS.Br, 2007).

Da norma ISO/IEC 12207, o MPS.Br recomenda que sejam utilizados os processos definidos, os quais podem ser adaptados seguindo o processo de

adaptação presente na norma; o MPS.Br baseia-se na ISO/IEC 15504 para a avaliação com dois objetivos: melhoria dos processos e determinação da capacidade de uma unidade organizacional avaliada; e do CMMI vieram os conceitos de maturidade e capacidade de processo (MPS.Br, 2007).

No MPS.Br, existem 7 níveis de maturidade, os quais são:

- ***Nível G***: o mais elementar, definido pelo modelo como *Parcialmente Gerenciado*. Inclui os processos de Gerência de Requisitos e Gerência de Projetos;
- ***Nível F***: definido pelo modelo como *Gerenciado*. Inclui os processos de Medição, Gerência de Configuração, Garantia de Qualidade, e Aquisição;
- ***Nível E***: definido pelo modelo como *Parcialmente Definido*. Inclui os processos de Definição do Processo Organizacional, Gerência de Reutilização, Gerência de Recursos Humanos, Gerência de Projetos (Evolução), e Avaliação e Melhoria do Processo Organizacional;
- ***Nível D***: definido pelo modelo como *Largamente Definido*. Inclui os processos de Desenvolvimento de Requisitos, Projeto e Construção de Protótipo, Integração de Produto, Verificação e Validação;
- ***Nível C***: definido pelo modelo como *Definido*. Inclui os processos de Gerência de Riscos, Desenvolvimento para Reutilização, Análise e Decisão e Resolução, e Gerência de Reutilização (evolução);
- ***Nível B***: definido pelo modelo como *Gerenciado Quantitativamente*. Inclui apenas uma evolução do processo de gerência de projeto;
- ***Nível A***: definido pelo modelo como *Em Otimização*. Inclui apenas o processo de Análise de Causas de Problemas e Resolução.

Esses níveis de maturidade são uma combinação dos processos junto com sua capacidade. A capacidade “é representada por um conjunto de atributos de processo (AP) descritos em termos de resultados esperados (RAP)” (MPS.Br, 2007, p. 16). A seguir é apresentada uma visão geral dos atributos de processo definidos no MPS.Br:

- **AP 1.1 O processo é executado**: indica apenas que o processo atingi o seu propósito;
- **AP 2.1 O processo é gerenciado**: indica que a execução do processo é gerenciada;
- **AP 2.2 Os produtos de trabalho do processo são gerenciados**: indica o quanto os produtos de trabalho produzidos pelo processo são gerenciados apropriadamente;
- **AP 3.1. O processo é definido**: indica o quanto o processo padrão da organização (ou unidade organizacional) é mantido para apoiar a implementação do processo definido;
- **AP 3.2 O processo está implementado**: indica o quanto o processo padrão é efetivamente implementado como processo definido;
- **AP 4.1 O processo é medido**: avalia se os resultados da medição são usados para assegurar o desempenho do processo no alcance dos seus objetivos;
- **AP 4.2 O processo é controlado**: indica o quanto o processo é estatisticamente controlado para produzir um processo estável;
- **AP 5.1 O processo é objeto de inovações**: indica que análises de causas de variação de desempenho ou até mesmo de causas inovadoras são utilizadas para identificar mudanças no processo;
- **AP 5.2 O processo é otimizado continuamente**: indica o quanto as mudanças na definição, gerência e desempenho do processo têm impacto efetivo para o alcance dos objetivos de melhoria do processo.

O Quadro 3 apresenta os níveis de maturidade do MPS.Br com os atributos de processo correspondentes a cada nível.

Nível	Processos	Atributos de Processo
A	Análise de Causas de Problemas e Resolução – ACP	AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP3.2, AP 4.1, AP 4.2 , AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI Desenvolvimento para Reutilização – DRU Análise de Decisão e Resolução – ADR Gerência de Reutilização – GRU (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
D	Verificação – VER Validação – VAL Integração do Produto – ITP Projeto e Construção do Produto – PCP Desenvolvimento de Requisitos – DRE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
E	Gerência de Projetos – GPR (evolução) Gerência de Reutilização – GRU Gerência de Recursos Humanos – GRH Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional – AMP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
F	Medição – MED Garantia da Qualidade – GQA Gerência de Configuração – GCO Aquisição – AQU	AP 1.1, AP 2.1 e AP 2.2
G	Gerência de Requisitos – GRE Gerência de Projetos – GPR	AP 1.1 e AP 2.1

Quadro 3 – Níveis de Maturidade no MPS.Br e respectivos Atributos de Processo.
Nota: Obtido em MPS.Br (2007)

Vale ressaltar que alguns atributos de processos possuem RAP’s que devem ser implementados ou não de acordo com o nível de maturidade desejado. Por exemplo, o AP2.1 possui 9 RAP’s esperados, porém apenas 8 devem ser implementados para o nível G, e um desses possui exigência diferente para o nível G e para o nível F (MPS.Br, 2007). Ainda segundo o MPS.Br (2007), "os atributos de processo AP4.1, AP 4.2, AP 5.1 e AP 5.2 somente devem ser implementados para os processos relevantes da organização/unidade organizacional. Os demais atributos de processo devem ser implementados para todos os processos”.

As duas principais diferenças em relação ao CMMI são: não possuir um modelo de representação contínua, apenas em estágios; e os níveis de capacitação serem mais gradativos. O nível 2 do CMMI foi dividido em dois e as

áreas de Planejamento de Projeto, e de Controle e Monitoramento de Projeto foram unidas em um único processo no MPS.Br, o processo de Gerência de Projeto. O nível 3 também foi dividido, mas em três níveis. Além disso, as áreas de processo Treinamento Organizacional, Gerência de Projetos Integrada, Foco no Processo Organizacional e Soluções Técnicas passaram a se chamar no MPS.Br, respectivamente, Gerência de Recursos Humanos, Gerência de Projeto (evolução), Avaliação e Melhoria do Processo Organizacional, e Projeto e Construção de Produto. Processos que abordam a reutilização de software também foram incluídos no MPS.Br, os quais são Gerência de Reutilização, Desenvolvimento para Reutilização e Gerência de Reutilização (evolução). Esses processos não são tratados no CMMI. O nível do MPS.Br correspondente ao nível 4 do CMMI não inclui as áreas de processo Desempenho do Processo Organizacional e Gerência Quantitativa de Projeto, que foram substituídas por Gerência de Projeto (evolução) e pelos atributos de processo AP4.1 e AP4.2. Finalmente, o nível mais alto do MPS.Br tem apenas o processo de Análise de Causas de Problemas e Resolução, excluindo a área de processo Distribuição e Inovação Organizacional do CMMI, que é substituída pelos atributos de processo AP5.1 e AP5.2.

Nas duas seções a seguir, os processos de gerência de requisitos e gerência de projetos do MPS.Br são discutidos em mais detalhes.

2.7 GERÊNCIA DE REQUISITOS

A gerência de requisitos, tanto no MPS.Br quanto no CMMI, é tratada como um processo, ou área de processo. Segundo ambos os modelos, o objetivo dessa área é “gerenciar os requisitos de produto e componentes de produto, e identificar inconsistências entre os requisitos” (MPS.Br, 2007, p. 26).

Já a norma ISO 12207 não contempla a gerência de requisitos como processo em separado. No Processo de Desenvolvimento, existem algumas tarefas relacionadas à gerência de requisitos. A atividade Análise dos Requisitos do Sistema, por exemplo, contém tarefas como avaliação dos requisitos de acordo

com critérios estabelecidos em contrato e rastreabilidade dos requisitos. A atividade de Planejamento, dentro do processo de Fornecimento, também tem tarefas relativas à gerência de requisitos, na medida em que inclui a revisão dos requisitos, de modo a elaborar uma estrutura para gerenciar o projeto.

O MPS.Br espera os seguintes resultados, quando o processo de gerência de requisitos encontra-se implementado em uma organização:

1. ***GRE1 – Entendimento dos Requisitos***: O entendimento dos requisitos é obtido junto aos fornecedores de requisitos. Corresponde no CMMI à prática *S.P 1.1 - Obter um Entendimento dos Requisitos*. Alguns dos produtos típicos de trabalho quando este resultado encontra-se implementado são aprovações dos requisitos junto aos fornecedores (CMMI, 2006);
2. ***GRE 2 – Aprovação de Requisitos***: Os requisitos de software são aprovados utilizando critérios objetivos. Corresponde no CMMI à prática S.P 1.1 (essa prática também envolve a aprovação dos requisitos) e à *SP 1.2 – Obter Comprometimento com os Requisitos*. Produtos de trabalho típicos são critérios de avaliação e aceitação dos requisitos e resultados de análise de acordo com esses critérios (CMMI, 2006);
3. ***GRE 3 – Rastreabilidade de Requisitos:*** A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida. Corresponse no CMMI à prática *SP1.4 – Manter Rastreabilidade bi-direcional nos requisitos*. Produtos de trabalho típicos aqui são matriz de rastreabilidade e sistema de rastreamento de requisitos (CMMI, 2006);
4. ***GRE 4 – Revisões de Requisitos***: Revisões em planos e produtos de trabalho do projeto são realizadas, visando identificar e corrigir inconsistências em relação aos requisitos. Corresponde no CMMI à prática *S.P 1.5 – Identificar Inconsistências entre Produtos de Trabalho e Requisitos*. Exemplos de produtos de trabalho são a

documentação de inconsistências (incluindo fontes, condições e raciocínio) e ações corretivas (SEI, 2006);

5. ***GRE 5 – Mudanças nos Requisitos***: Mudanças nos requisitos são gerenciadas ao longo do projeto. Corresponde no CMMI à prática *SP 1.3 – Gerenciar Alterações de Requisitos*. Status dos requisitos, e decisões em cima de alterações são alguns dos produtos de trabalho típicos (SEI, 2006).

2.8 GERÊNCIA DE PROJETO

No CMMI, as áreas de processo Planejamento de Projeto (PP), e Controle e Monitoramento de Projeto (PMC) correspondem ao processo de Gerência de Projeto do MPS.Br, cujo objetivo "é estabelecer e manter planos que definam 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." (MPS.Br, 2007, p. 23).

Já a norma ISO 12207, o planejamento do projeto, bem como seu controle e monitoramento, estão inseridos no Processo de Fornecimento, através das atividades de Planejamento, e de Execução e Controle.

Para o nível G, o MPS.Br espera os seguintes resultados, quando o processo de gerência de projeto encontra-se implementado em uma organização:

1. ***GPR 1. O escopo do trabalho para o projeto é definido***: corresponde no CMMI à prática *S.P 1.1 – Estimar o Escopo do Projeto*, da área de processo PP. Alguns dos produtos típicos de trabalho são as descrições das tarefas e uma Estrutura Analítica de Projeto (EAP), chamada no CMMI de *Work Breakdown Structure* (WBS) (SEI, 2006);
2. ***GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados***: os principais produtos de trabalho e tarefas do projeto são decompostos em

componentes menores, mais facilmente gerenciáveis e possíveis de serem dimensionados. Corresponde no CMMI à prática *SP1.2 – Estabelecer Estimativas para Tarefas e Produtos de Trabalho* da área de processo PP. Tem como produtos de trabalho típicos modelos de estimativas, tamanho e complexidade de tarefas e produtos de trabalho (SEI, 2006);

3. ***GPR 3. O modelo e as fases do ciclo de vida do projeto são definidas***: consiste na definição de fases, ao final das quais podem ser feitas avaliações e/ou decisões para o projeto. Corresponde no CMMI à prática *S.P 1.3 – Definir Ciclo de Vida do Projeto* da área de processo PP; e tem as fases do ciclo de vida do projeto como produto de trabalho típico;
4. ***GPR 4. O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas:*** estimativas de esforço e custo são realizadas com base em resultados de análises utilizando dados históricos e/ou referências técnicas. Corresponse à prática *S.P 1.4 – Definir Estimativas de Esforço e Custo* da área de processo PP, e tem como produtos típicos o raciocínio de estimativas, e as próprias estimativas de esforço e custo;
5. ***GPR 5. O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos***: orçamento e cronograma são estabelecidos baseados nas estimativas realizadas e monitoradas durante o andamento do projeto. Corresponde no CMMI às práticas *S.P 2.1 – Estabelecer Cronograma e Orçamento,* da área de processo PP, e *S.P 1.1 – Monitorar Parâmetros de Planejamento de Projeto*, da área de processo PMC. Tem como produtos típicos, além do cronograma e orçamento, registro de performance do projeto;
6. ***GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são***

determinados e documentados: riscos que podem ocorrer no projeto são identificados, analisados, priorizados e documentados, bem como planos de mitigação são elaborados. Corresponde no CMMI às práticas *S.P 2.2 – Identificar Riscos do Projeto,* da área de processo PP, e *S.P 1.3 – Monitorar Riscos do Projeto*, da área de processo PMC. Exemplos de produtos de trabalho típicos são os riscos identificados e os registros de monitoramento dos riscos;

7. ***GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e os conhecimentos necessários para executá-lo***: as funções, responsabildades e relações hierárquicas para o projeto são determinadas. Corresponde no CMMI à prática *S.P 2.5 – Planejar Conhecimentos e Habilidades Necessárias* da área de processo PP*,* e tem a base de treinamento e conhecimentos dos recursos humanos da organização como um dos produtos de traballho típicos;
8. ***GPR 8. As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados***: a quantidade necessária de recursos (maquinário, material, etc) para executar as atividades do projeto é planejada com base nas estimativas iniciais. Corresponde no CMMI à prática *S.P. 2.4 – Planejar Recursos do Projeto*, da área de processo PP, e tem como exemplo de produto de trabalho típico a lista de equipamentos necessários;
9. ***GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição***: todos os dados do projeto (atas de reunião, relatórios, códigos fonte) são planejados, incluindo a forma como serão distribuídos. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança. Além disso, monitoramento de dados são realizados em marcos e/ou pontos de controle do projeto. Corresponde no CMMI às práticas *S.P*

2.3 – Planejar o Gerenciamento de Dados, da área de processo PP, e *S.P 1.4 – Monitorar Gerenciamento dos Dados,* da área de processo PMC. Tem a lista de dados, os mecanismos de coleta e armazenamento, e os requisitos de privacidade e segurança como alguns produtos de trabalho típicos;

10. ***GPR 10. (Até o nível F). Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto***: todas as informações coletadas para o projeto são reunidas em um único documento, o plano de projeto, que é o produto de trabalho típico. Corresponde no CMMI à prática *S.P 2.7 – Estabelecer o Plano de Projeto* da área de processo PP;
11. ***GPR 11. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada***: aspectos técnicos, financeiros e humanos são analisados para verificar se o projeto irá atingir as metas. Se necessário, ajustes são realizados para atender as metas. Corresponde no CMMI às práticas *S.P 3.1 – Revisar Planos que afetam o Projeto* e *S.P 3.2 – Conciliar níveis de Recurso e Trabalho* da área de processo PP. Tem como alguns dos produtos de trabalho típicos o registro das análises, cronograma revisado e custo renegociado;
12. ***GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido***: a obtenção desse compromisso involve a interação tanto com os interessados internos quanto externos ao projeto. Corresponde no CMMI às práticas *S.P 3.1 – Revisar Planos que afetam o Projeto* e *3.3 – Obter Comprometimento com o Plano* da área de processo PP. Os documentos de solicitação de compromisso e de compromisso propriamente dito são alguns dos produtos de trabalho típicos;
13. ***GPR 13. O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados***: a aderência aos diversos planos deve ser avaliada

continuamente, durante todo o projeto. Corresponde no CMMI às práticas *S.P 1.1 – Monitorar Parâmetros de Planejamento de Projeto* e *S.P1.6 – Conduzir Revisões de Progresso*, ambos da área de processo PMC. Os registros de desempenho do projeto e de desvios significativos representam alguns produtos de trabalho típicos;

14. ***GPR 14. O envolvimento das partes interessadas no projeto é gerenciado***: todo o envolvimento dos interessados deve ser acompanhado conforme o previsto no plano. Corresponde às práticas *S.P 1.2 – Monitorar Compromissos* e *S.P 1.5 – Monitorar Envolvimento de Interessados*, ambas da área de processo PMC.. Registros de revisões de compromissos são um dos produtos típicos de trabalho;
15. ***GPR 15. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento***: Diferente do GPR 13, esas revisões são realizadas em marcos definidos no planejamento do projeto, e não no dia a dia. Refere-se no CMMI à prática *S.P 1.7 – Conduzir Revisões de Marcos*, da área de processo PMC. Registros de revisões indicam que o resultado está implementado;
16. ***GPR 16. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas***: as revisões diárias e de marco podem indicar uma lista de problemas que estão ocorrendo no projeto, e que devem ser identificadas e analisadas. Refere-se no CMMI à prática *S.P 2.1 – Analisar Questões* da área de processo PMC. A lista de problemas ou questões é um dos produtos de trabalho típicos;
17. ***GPR 17. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão***: Refere-se no CMMI à prática *S.P. 2.2 Tomar Ações*

Corretivas, da área de processo PMC, e tem como produto de trabalho típico os planos de ação corretiva.

2.9 CONCLUSÕES

Neste capítulo foram apresentadas normas internacionais referentes à qualidade de produto e a processos de ciclo de vida de software. Foram também apresentados dois modelos, um nacional e outro internacional, referente à capacitação de processos de sofware.

O conteúdo deste capítulo é de fundamental importância para o entendimento da infra-estrutura proposta. A ISO 9126 forneceu as características a serem consideradas no processo elaborado para avaliação e seleção de ferramentas CASE, como também forneceu um conjunto de métricas, as quais foram selecionadas e adaptadas para o propósito de integração de ferramentas. A ISO/IEC 14598-5: Guia para Avaliação foi utilizada como base para a definição do processo de avaliação e seleção de ferramentas CASE. O CMMI e a ISO 12207 foram utilizados para definir os requisitos funcionais e de informacao das ferramentas selecionadas para apoio ao nível G do MPS.Br, pois o MPS.Br apenas defini os resultados esperados, já que não tem como objetivo reproduzir o conteúdo de outras normas e modelos.

CAPÍTULO 3. ONTOLOGIAS

Neste capítulo, é apresentada uma visão geral sobre ontologias, incluindo seus elementos, algumas classificações propostas, principais linguagens de representação, métodos para a elaboração, além de critérios a serem utilizados na avaliação.

3.1 VISÃO GERAL

O termo ontologia tem origem nos termos gregos "ontos" e “logos”, que significam “ser” e “palavra”, respectivamente (CHANDRASEKARAN; JOSEPHSON; BENJAMINS, 1999), tendo, na Filosofia, o objetivo conceituar tudo aquilo que pode “ser” ou “existir”. Ontologias, no entanto, têm sido adotadas em diferentes áreas de conhecimento ao longo do tempo. Segundo uma definição amplamente aceita na Inteligência Artificial, “uma ontologia é uma especificação de uma conceituação, isto é, uma descrição de conceitos e relações que podem existir para um agente ou uma comunidade de agentes.” (GRUBER, 1995, p. 2-3). Guarino (1995) define ontologia como uma descrição parcial e explícita de uma conceituação, enquanto Fikes e Farquhar (1999) definem ontologia como sendo uma teoria sobre um domínio, na qual deve ser especificado um vocabulário de entidades, classes, propriedades, predicados e funções, além de um conjunto de relações que necessariamente amarram esse vocabulário.

Em computação, o maior objetivo da ontologia está em atribuir sentido e significado para determinados termos em um certo contexto. Uschold e Gruninger (1996) mostram algumas das principais aplicações para ontologias:

- ***Comunicação entre pessoas e organização***: pessoas, organizações, sistemas de software ou outros agentes quaisquer podem realizar qualquer tipo de comunicação. No entanto, tais agentes estão inseridos em diferentes contextos e, conseqüentemente, têm diferentes pontos de vista para um mesmo item. A ontologia define conceituações de modo que a

comunicação entre pessoas com pontos de vista diferentes seja feita utilizando uma terminologia comum. Segundo Oliveira (1999, p. 37), “ontologias capacitam o entendimento compartilhado e a comunicação entre pessoas com diferentes necessidades e pontos de vista particulares sobre o seu trabalho”;

- ***Interoperabilidade entre sistemas***: refere-se à necessidade de diferentes usuários utilizarem diferentes ferramentas ou compartilhar dados. As ontologias ajudam no sentido de que pessoas e ferramentas passam a ter a mesma visão de um determinado conceito. Essa interoperabilidade pode ser assegurada pela “tradução” dos conceitos entre os diversos métodos de modelagem, paradigmas, linguagens e ferramentas de software (USCHOLD; GRUNINGER, 1996). Por exemplo, na comunicação entre *n* linguagens, ao invés de se utilizar um tradutor para cada par de linguagens, basta utilizar um tradutor de cada linguagem para a ontologia, e desta para a linguagem específica. Neste caso, a ontologia funciona como uma interlíngua;
- ***Engenharia de sistemas***: em particular, em três aspectos: *re-usabilidade,* uma vez que o conhecimento compartilhado entre domínios de interesse, uma vez capturado em uma ontologia, pode promover a reutilização; *especificação de requisitos*, visto que o conhecimento compartilhado auxilia o processo de elicitação de requisitos, no sentido de facilitar a identificação e explicar os conceitos envolvidos nos requisitos de um sistema; e *confiabilidade*, já que ajudam na verificação da especificação do software.

Falbo e Duarte (2000) discutem ainda outras vantagens, também voltadas para a área de Tecnologia de Informação, para a elaboração de ontologias, as quais são: ***compreensão de áreas de conhecimento***, pois, para desenvolver uma ontologia, as pessoas envolvidas devem explicar detalhadamente o entendimento do domínio, o que ajuda a melhorar a compreensão do mesmo; e ***consenso sobre o entendimento de áreas de conhecimento***, pois diferentes especialistas podem ter um entendimento diferenciado sobre os conceitos

envolvidos e, na construção da ontologia, as diferenças são explicitadas, objetivando atingir um consenso sobre o significado e a importância deles.

3.2 ELEMENTOS DE UMA ONTOLOGIA

Segundo Uschold e Gruninger (1996), as ontologias devem necessariamente prover um vocabulário de termos e alguma especificação de seu significado, ou seja, as suas definições. Para Falbo e Duarte (2000), uma ontologia consiste basicamente dos conceitos e relações que existem em um determinado domínio, suas definições e propriedades, além de restrições descritas na forma de axiomas.

Chandrasekaran, Josephson e Benjamins (1999) afirmam que uma ontologia deve possuir os seguintes itens:

- ***Objetos***, que são os termos que a ontologia deve proporcionar;
- ***Propriedades*** ou ***Atributos***, que assumem valores e ajudam a explicar o conceito do objeto;
- ***Relacionamentos*** entre os vários objetos;
- ***Eventos***, que podem ocorrer em determinados instantes em um dado objeto;
- ***Processos***, dos quais os objetos da ontologia participam;
- ***Estados*** em que os objetos podem se encontrar;
- ***Efeitos*** causados pelos eventos, os quais podem ser outros eventos ou gerar novos estados.

Ainda segundo o autor, a representação dos objetos, seus relacionamentos, estados, eventos e processos deve estar necessariamente inserida em um determinado domínio, sem o qual esses itens não têm sentido. Por exemplo, modelar os objetos *Animal*, *Mineral* e *Planta* não teria sentido algum sem um domínio específico. Porém, ao colocar tais objetos no domínio *Planeta Terra*, tais objetos passam a ter sentido como sendo os seus elementos constituintes. Atributos podem ser definidos para os conceitos a fim de caracterizá-los. Por

exemplo, podem ser definidos os atributos *Possui_Vida(x)* para Animal e Planta, ou *Contem_Carbono(x)*, informando se Animal, Mineral ou Planta é composto por carbono. Como exemplo de relacionamento, tem-se *Pode_Comer(x,y)*, informando que o elemento x (por ex., Animal) pode comer o elemento y (por ex, Planta) (CHANDRASEKARAN; JOSEPHSON; BENJAMINS, 1999). Os objetos, propriedades e relacionamentos são específicos desse domínio e devem necessariamente refletir essa realidade.

3.3 CLASSIFICAÇÃO DE ONTOLOGIAS

Existem diversas propostas de classificação de ontologias. Oliveira (1999) apud Heijst (1997) e Heijst (1995) fornece duas dimensões para a classificação de ontologias: a primeira é a quantidade e o tipo de estrutura da conceituação e a outra é o assunto da conceituação. Na primeira dimensão estão:

a) **Ontologias Terminológicas**, que definem termos utilizados para representar o conhecimento no domínio de um discurso;
b) **Ontologias de Informação:** que especificam a estrutura de registro em um esquema de banco de dados; e
c) **Ontologias de Modelagem de Conhecimento:** que especificam a conceituação do conhecimento, possuindo uma estrutura interna mais rica se comparada com as ontologias de informação. São frequentemente ajustadas a um uso específico do conhecimento.

Já a segunda dimensão é dividida em quatro categorias, as quais são:

a) **Ontologias Genéricas:** descrevem conceitos genéricos, que não dependem de um domínio ou problema em particular, ou seja, podem ser utilizados em várias áreas. Alguns exemplos são espaço, tempo, processo, matéria, componente, dentre outros;
b) **Ontologias de Domínio:** fornecem vocabulários que se referem a um determinado domínio. Pode ser utilizado tanto a especialização de conceitos presentes na ontologia genérica quanto restrições na estruturação dos conceitos;

c) **Ontologias de Aplicação:** possuem as definições necessárias para modelar um conhecimento em um domínio, e são uma combinação de conceitos tirados das duas categorias anteriores;
d) **Ontologias de Representação:** referem-se às conceituações que fundamentam os formalismos para representação de conhecimento. Estas são neutras em relação às entidades de um domínio. Segundo (SANTIN et al, 2009), aplicações de comércio eletrônico e bibliotecas virtuais são fortemente baseadas em ontologias de representação.

Guarino (1998) propõe que ontologias sejam classificadas segundo outras duas dimensões: nível de detalhe e nível de dependência. Na dimensão do nível de detalhe, a ontologia é construída para estabelecer consenso sobre o compartilhamento de um vocabulário (conceituação) ou de uma base de conhecimento que usa esse vocabulário (OLIVEIRA, 1999). Diferente da dimensão do nível de dependência, não são mencionadas categorias.

Na dimensão do nível de dependência, Guarino (1998) considera as ontologias de alto nível, de domínio, de tarefas e de aplicação, conforme ilustra a Figura 4. A diferença aqui é a exclusão das ontologias de representação e inclusão das ontologias de tarefas. Nessa classificação, as ontologias de **alto nível** descrevem os conceitos mais genéricos, como tempo, espaço, objetos, eventos, ações, e servem geralmente de base para as ontologias mais específicas. Os conceitos que elas representam são tipicamente independentes de um problema em particular. As ontologias de **tarefas** têm o objetivo de descrever vocabulários relacionados a tarefas ou atividades genéricas, através da especialização de conceitos introduzidos nas ontologias de alto nível. As ontologias de **domínio** descrevem o vocabulário relacionado a um domínio específico, como medicina ou automobilístico, por exemplo. Por fim, as ontologias de **aplicação** são as mais específicas, usadas dentro de aplicações e podem especializar tanto conceitos das ontologias de tarefa quanto das ontologias de domínio (GUARINO, 1995). Desta forma, Guarino (1998) propõe que ontologias sejam construídas segundo seu nível de dependência. Por exemplo, os conceitos

da ontologia de aplicação seriam especializações das ontologias de domínio e de tarefas, que por sua vez seriam especializações das ontologias genéricas.

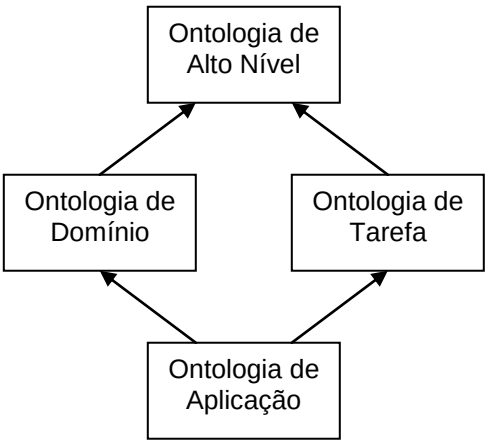


Figura 4 – Tipos de Ontologias
Nota: Obtido em GUARINO (1998).

3.4 LINGUAGENS PARA A REPRESENTAÇÃO DE ONTOLOGIAS

Um aspecto que deve ser levado em consideração na construção de ontologias são as linguagens de representação. Segundo Oliveira (1999), as linguagens são “um ponto importante para operacionalizar a ontologia, ou seja, permitir sua efetiva utilização”. Algumas das linguagens mais citadas para a representação de ontologias são:

a) **Lógica de Primeira Ordem:** por ser geral, bem conhecida e por adicionar poucos compromissos ontológicos.É bastante utilizada para representar ontologias (VALENTE, 1995);
b) **KIF (*Knowledge Interchange Format*):** linguagem projetada para a troca de conhecimento entre sistemas distintos. Torna explícita as decisões relativas à representação do conhecimento e permite que o usuário defina novas construções sem mudar a linguagem (GENESERETH, 2009);
c) **Ontolingua:** que foi construída baseada na linguagem KIF e inclui mecanismos para a definição de classes e relações. É uma linguagem

formal que foi projetada com o propósito específico de expressar ontologias (GRUBER, 1992);

d) ***Description Logic*:** essa linguagem tem lógica voltada para a definição de categorias. Tem como principais mecanismos de inferência verificar se uma categoria é um sub-conjunto de outra ou se um objeto pertence a uma categoria (RUSSEL; NORVIG, 1995);

e) ***LINGO*:** uma linguagem gráfica que tem sua representação formal explicitamente definida em lógica de 1ª Ordem (FALBO, 1998);

f) ***UML* (*Unified Modeling Language*):** segundo Mian (2003), a UML é apropriada para a representação gráfica de ontologias devido ao seu suporte computacional e à ampla comunidade de usuários. Um exemplo de ontologia representada em UML pode ser encontrada em Kitchenham et al (1999);

g) ***OWL:*** linguagem utilizada para definir e instanciar ontologias na WEB. Além de incluir descrições de classes, suas propriedades, seus relacionamentos e restrições, também permite a instanciação de ontologias. Essa linguagem é a recomendada pela W3C (*World Wide Web Consortium*) e possui 3 sub-linguagens: *OWL LITE*, *OWL DL* e *OWL Full* (W3C, 2009).

3.5 MÉTODOS PARA A ELABORAÇÃO DE ONTOLOGIAS

Segundo Oliveira (1999), os métodos para a elaboração de ontologias propostos na literatura surgiram no contexto de projetos específicos. “Cada grupo segue um conjunto de princípios, critérios de projeto e fases para o desenvolvimento de ontologias”. Alguns métodos são descritos a seguir.

Grüninger e Fox (1995) propuseram uma metodologia para projeto e avaliação de ontologias dentro do projeto TOVE (*Toronto Virtual Enterprise*), derivada da experiência na elaboração de ontologias para os domínios de processos corporativos e de negócios. Suas etapas são a *Descrição dos cenários de motivação*, a *Definição de questões informais de competência*, a *Especificação*

da terminologia em lógica de primeira ordem, a *Definição formal das questões de competência*, uma *Especificação de axiomas em lógica de primeira ordem* e a *Avaliação da ontologia*. Na *Descrição dos cenários de motivação*, são descritos os cenários de uso que motivam a elaboração da ontologia. A *Definição de questões informais de competência* é feita com base nos cenários de motivação e representam os requisitos da ontologia, devendo ser usadas para avaliar se os compromissos da ontologia são necessários e suficientes para caracterizar os problemas dos cenários de motivação e suas soluções" (FALBO, 1998, p. 68). A *Especificação da terminologia em lógica de primeira ordem* é feita com base nas questões informais de competência elaboradas. A *Definição formal das questões de competência* é feita com base nas questões informais e na especificação da ontologia em lógica de primeira ordem. Na *Especificação de axiomas em lógica de primeira ordem*, axiomas são definidos como sentenças de primeira ordem usando os predicados da ontologia. Por fim, a *Avaliação da ontologia* é feita com base nas questões formais de competência, que são usadas para avaliar se o conjunto de axiomas está completo.

O método proposto por Uschold e Gruninger (1996) apresenta quatro etapas. A primeira é *Identificação do Escopo e da Proposta*, onde é necessário identificar o porque de a ontologia estar sendo construída e seu uso pretendido. Se for o caso, pode-se também informar o público-alvo. A segunda etapa, *Construção da Ontologia*, encontra-se dividida em: *captura* (identificação dos conceitos chaves e seus relacionamentos, seguida da elaboração das definições), *codificação* (representação explícita da conceitualização, que compreende o compromisso com os conceitos básicos, a escolha de uma linguagem para a representação e a codificação) e *integração com ontologias existentes* (utilização, em todo ou em parte, de ontologias já pré-existentes). A terceira etapa é a *Avaliação*, que consiste na avaliação da ontologia elaborada com base em especificações, questões de competência e/ou comparação com o mundo real. A quarta e última etapa é a *Documentação*, que consiste na elaboração de documentos sobre os principais conceitos definidos e as primitivas utilizadas para expressar as definições.

Férnandez, Gómez-Pérez e Juristo (1997) propuseram um método estruturado para a elaboração de ontologias, chamado de *METHONTOLOGY*. Esse método tem três etapas básicas. A primeira é a *Especificação*, cujo objetivo é produzir um documento de especificação da ontologia. Esse documento contém o seu propósito (inclusive potenciais usuários e cenários de uso), o nível de formalidade da ontologia a ser implementada e o escopo (com conjunto de termos a serem representados, suas características e granularidade). A segunda etapa corresponde à *Conceituação*. Segundo Falbo (1998), essa é a parte mais trabalhosa, "pois envolve a elaboração de várias representações intermediárias utilizadas para conceituar o domínio, tais como Dicionário de Dados, Árvores de Classificação de Conceitos, Tabela de Constantes, Tabelas de Atributos de Instâncias e de Classe, Tabelas de Fórmulas, Árvores de Classificação de Atributos, e Tabela de Instâncias". Para acelerar essa etapa e, conseqüentemente, o processo de elaboração da ontologia, o próprio método propõe a reutilização de ontologias que forneçam definições para os termos identificados, cuja semântica e implementação sejam coerentes. A terceira etapa é a *Implementação*, cujo resultado é a ontologia codificada em uma linguagem formal. Por fim, tem-se a *Avaliação*, que deve ser realizada em cada etapa e cujo objetivo é avaliar a ontologia com base na especificação de requisitos.

Já Falbo (1998) reuniu as melhores características desses métodos e elaborou outro, com as seguintes etapas: *Propósito e Especificação de Requisitos*, que tem como objetivo identificar a competência da ontologia; *Captura da Ontologia*, na qual devem ser identificados os conceitos e as relações relevantes, bem como devem ser definidos os axiomas que especificam as definições dos termos da ontologia e as restrições sobre sua interpretação; *Formalização da Ontologia*, que é a representação da ontologia em linguagem formal, visto que a linguagem natural pode dar origem a incertezas e ambigüidades; *Integração de Ontologias Existentes*, que tem o objetivo de integrar a ontologia com outras ontologias existentes, visando aproveitar conceituações previamente definidas; *Avaliação da Ontologia*, quando deve ser avaliado se a ontologia satisfaz os requisitos estabelecidos na primeira etapa; e *Documentação da Ontologia*, na qual

devem ser incluídos os propósitos, requisitos, descrições textuais da conceituação, ontologia formal e critérios de projeto.

O Quadro 4 apresenta um resumo dos métodos citados, comparando as etapas propostas com as quatro etapas básicas mencionadas em (USCHOLD, 1996), as quais são:

1. **Definição do Propósito**: o objetivo é motivar a construção da ontologia, informando o porquê de ela ser construída, ou seja, a sua utilização (compartilhamento, especificação, integração, etc.);
2. **Construção**: são capturadas as definições da ontologia e formalizadas em alguma linguagem específica. Existe ainda a possibilidade de integração com ontologias existentes;
3. **Avaliação**: consiste na avaliação da ontologia com base em critérios previamente especificados;
4. **Documentação**: refere-se à documentação dos conceitos e axiomas identificados.

Proponente	Etapa Proposta	Etapa de Referência
GRUNINGER; FOX, 1995	*Identificação do Escopo e da Proposta*	Definição do Propósito
	Construção da Ontologia	Construção
	Avaliação	Avaliação
	Documentação	Documentação
USCHOLD; GRUNINGER, 1996	*Descrição dos cenários de motivação*	Definição do Propósito
	Definição de questões informais de competência	
	Especificação da terminologia em lógica de primeira ordem	Construção
	Definição formal das questões de competência	
	Especificação de axiomas em lógica de primeira ordem	
	Avaliação	Avaliação
FÉRNANDEZ; GÓMEZ-PÉREZ; JURISTO, 1997	*Especificação*	Definição do Propósito
	Conceituação	Construção
	Implementação	
	Avaliação	Avaliação
FALBO, 1998	*Propósito e Especificação de Requisitos*	Definição do Propósito
	Captura da Ontologia	Construção
	Formalização da Ontologia	
	Integração de Ontologias Existentes	
	Avaliação da Ontologia	Avaliação
	Documentação da Ontologia	Documentação

Quadro 4 – Metodologias para Elaboração de Ontologias

As etapas definidas em cada uma das propostas acima não devem ser consideradas seqüenciais, e sim parte de um processo iterativo. Falbo (1998) afirma, por exemplo, que a etapa de captura pode apontar requisitos que não foram previamente identificados, ou, na avaliação, pode-se perceber que os termos descritos não são suficientes para conceituar o propósito especificado. Do mesmo modo, incoerências podem ser detectadas, provocando uma revisão das especificações e dos termos definidos na ontologia.

3.6 CRITÉRIOS DE QUALIDADE PARA A ELABORAÇÃO DE ONTOLOGIAS

Todos os métodos de elaboração de ontologias apresentados incluem a avaliação da ontologia desenvolvida. Segundo Falbo (1998), essa avaliação deve ser realizada com base em critérios de qualidade objetivos, fundamentados no propósito do artefato resultante. Uschold e Gruninger (1996) enumeraram alguns critérios para a avaliação de ontologias, que devem ser usados para guiar o processo de elaboração da ontologia em todas as suas etapas. Os critérios especificados por ele foram:

- **Clareza**: nessa critério é avaliada a objetividade das definições, de forma a minimizar a ambigüidade. Uschold e Gruninger (1996) sugerem que sejam dados exemplos para ajudar a entender as definições elaboradas;
- **Consistência**: a ontologia deve ser consistente, tanto na definição dos axiomas quanto na linguagem natural;
- **Extensibilidade**: refere-se à definição de novos termos para usos específicos, sem que as definições existentes tenham que ser revistas. Os dois critérios a seguir ajudam a assegurar a extensibilidade;
- **Compromissos de Codificação Mínimos**: a conceituação não deve ser especificada com base em uma tecnologia particular de representação de conhecimento;

- **Compromissos Ontológicos Mínimos**: o mínimo possível de compromissos ontológicos devem ser estabelecidos, de forma a permitir que as partes comprometidas com a ontologia possam especializá-la e instanciá-la conforme necessário.

Fox, Chionglo e Fadel (1993) propõem ainda outro critério de qualidade, o qual dá ênfase às questões de competência. As questões de competência representam as questões que a ontologia pode responder ou as tarefas que ela pode suportar, sendo utilizadas, portanto, para determinar a capacidade da ontologia de resolver problemas. Essas questões, identificadas para definir o propósito da ontologia, são utilizadas para avaliar se a ontologia atende ao propósito pretendido.

3.7 ATUAIS APLICAÇÕES NA ENGENHARIA DE SOFTWARE

Algumas ontologias têm sido elaboradas para o domínio de engenharia de software, dentre elas estão a ontologia de Qualidade de Software proposta por Falbo e Duarte (2000), as ontologias de Manutenção de Software elaboradas por Dias (2003) e por Kitchenham (1999), e a ontologia de Processos de Software definida por Bertollo e Falbo (2005).

A ontologia de Qualidade de Software, representada em LINGO, foi definida com o objetivo de fornecer um vocabulário comum para o domínio de qualidade de software. Ela responde a questões sobre características de qualidade de um determinado artefato, quais características podem ser medidas e métricas a serem utilizadas, dentre outras questões (FALBO; DUARTE, 2000).

A ontologia de Manutenção de Software proposta por Dias (2003), também representada em LINGO, foi elaborada com o objetivo de obter consenso e fornecer conhecimento sobre o vocabulário utilizado na manutenção de software. A ontologia inclui termos sobre o domínio da aplicação e os procedimentos de manutenção de software utilizados na organização, e até mesmo sobre o próprio sistema e sobre a linguagem de programação e os métodos de desenvolvimento

utilizados. Essa ontologia foi definida como base para a construção de um sistema de gerência de conhecimento para manutenção (DIAS, 2003). Já a ontologia elaborada por Kitchenham (1999) foi elaborada com o objetivo de prover um *framework* falando sobre o domínio da manutenção e identificar fatores que influenciam na manutenção (KITCHENHAM, 1999).

A ontologia de Processos de Software foi graficamente representada utilizando UML e tem como objetivo fornecer a compreensão, por parte das organizações de desenvolvimento de software, dos conceitos relacionados com processos de software, incluindo padrões e metodologias (BERTOLLO; FALBO, 2005).

3.8 CONCLUSÕES

Este capítulo foi dedicado ao tema ontologias, destacando suas aplicações, elementos, algumas classificações que têm sido propostas, linguagens para representação, métodos para a sua elaboração, além de critérios para avaliação. Adicionalmente, são citadas algumas ontologias já elaboradas na área de engenharia de software.

O método para elaboração de ontologias proposto por Falbo (1998) foi utilizado na elaboração das ontologias de Gerência de Projeto e Gerência de Requisitos apresentadas no capitulo 5, tendo sido a primeira etapa executada conforme proposto por Grüninger e Fox (1995) . O intuito foi obter um conjunto de ontologias que apóiem a integração de ferramentas que facilitem a execução dos processos pertinentes ao nível G do MPS.Br. Tal conjunto de ontologias possuem dupla aplicação no contexto: 1) prover um entendimento compartilhado dos requisitos de informação das ferramentas a serem integradas e 2) funcionar como interlíngua, apoiando a tradução dos termos específicos, utilizados por cada ferramenta, nos termos definidos pelas ontologias.

CAPÍTULO 4. PROCESSO PARA AVALIAÇÃO E SELEÇÃO DE FERRAMENTAS DE SOFTWARE

Neste capítulo, é apresentado o processo, definido como parte desta dissertação, para avaliação e seleção de ferramentas CASE. Para demonstrar a sua adequação, o processo foi utilizado para selecionar ferramentas livres que apóiam a obtenção dos resultados esperados do nível G do MPS.Br.

4.1 VISÃO GERAL

Uma das questões tratadas na Engenharia de Software é a disponibilização de ferramentas para o desenvolvimento e manutenção de software bem como para a gerência desses processos, de forma que os produtos possam ter melhor qualidade. Devido à complexidade e quantidade das atividades a serem executadas, várias ferramentas são necessárias. Para promover o sucesso do desenvolvimento e da manutenção de sistemas de software em uma organização, atenção especial deve ser dada à seleção de tais ferramentas, chamadas ferramentas CASE (*Computer-Aided Software Engineering*), tanto com relação aos requisitos técnicos quanto aos gerenciais (ISO/IEC, 1995).

Ferramentas CASE livres são uma alternativa interessante para pequenas e médias empresas reduzirem os custos dos processos de software. No entanto, muitas dessas ferramentas possuem características não desejáveis, como interface com usuário inconsistente, e instalação e configuração difíceis. Essas características são consequência da maneira como as ferramentas livres são, em geral, desenvolvidas e de sua própria natureza. Neste contexto, é fundamental dispor de um processo para apoiar a seleção do melhor conjunto de ferramentas livres para atender a um objetivo específico.

A ISO/IEC 14102 (*Tecnologia de Informação: Guia para Avaliação e Seleção de Ferramentas CASE*) é dedicada exclusivamente ao processo de avaliação e

seleção de ferramentas CASE, definindo um conjunto de processos e de características que devem ser considerados neste contexto. O conjunto de características é uma extensão das características definidas na ISO/IEC 9126, tendo em vista as particularidades desejadas em ferramentas CASE. A norma orienta como identificar os requisitos organizacionais, mapear esses requisitos para avaliação das características das ferramentas CASE, além de prover um processo para seleção da ferramenta CASE mais apropriada, baseado na medição das características definidas (ISO/IEC, 1995).

Essa norma, junto com as normas ISO/IEC 14598-5 e ISO/IEC 9126 foram utilizadas e adaptadas para elaborar o processo de avaliação e seleção de ferramentas CASE descrito abaixo.

4.2 DESCRIÇÃO DO PROCESSO

A Figura 5 apresenta as macro-atividades do processo. A linguagem utilizada para representação gráfica do processo foi proposta por Villela (2004) e encontra-se descrita no Anexo A.

Figura 5 – Macro-Atividades do Processo de Seleção

Macro-Atividade 1 - Iniciação

A proposta desta macro-atividade é definir os objetivos e requisitos gerais para a avaliação e seleção de ferramentas CASE, de forma a orientar a execução do processo e permitir a definição dos aspectos necessários a gerência do mesmo (ex.: custo, cronograma e recursos) (ISO/IEC, 1995). Suas atividades, que foram adaptadas das normas ISO/IEC 14102 e ISO/IEC 14598, são apresentadas na Figura 6 e detalhadas em seguida.

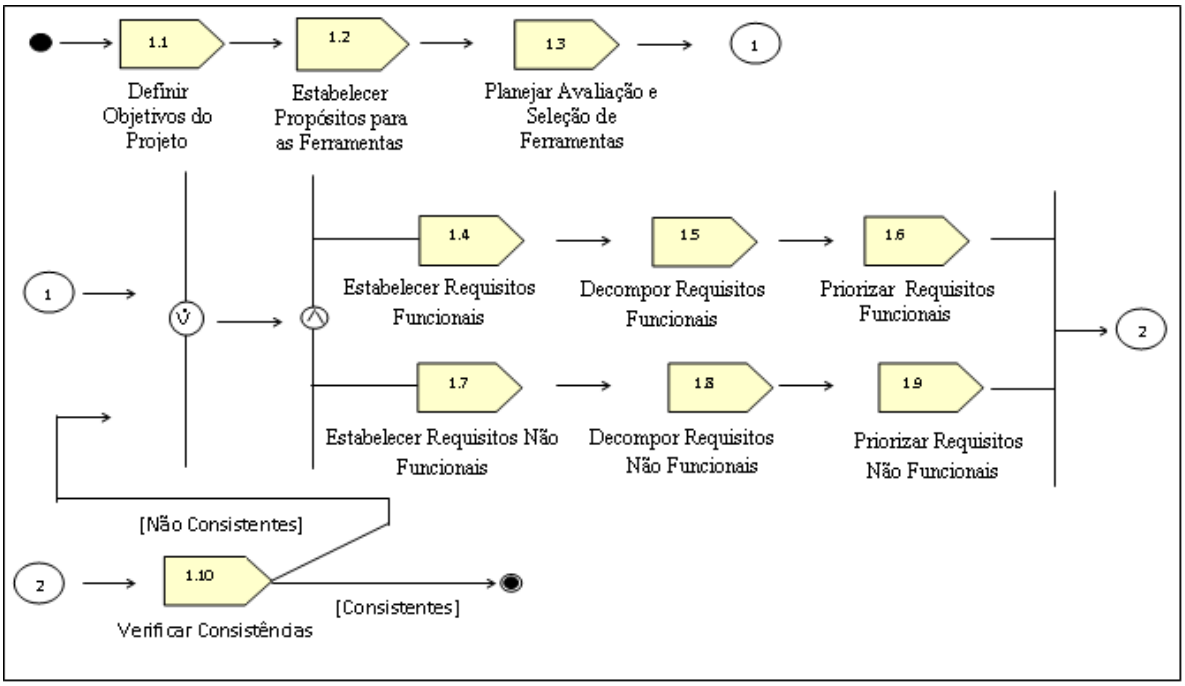


Figura 6 – Atividades de Iniciação

Atividade 1.1 - Definir Objetivos do Projeto: resume-se em definir os objetivos gerais do projeto;

- Entrada(s): Não existem entradas para essa atividade;
- Saída(s): Objetivos gerais do projeto.

Atividade 1.2 - Estabelecer Propósitos para as Ferramentas: refere-se à decomposição dos objetivos até um nível que seja possível estabelecer os propósitos a serem atingidos pelas ferramentas;

- Entrada(s): Objetivos gerais do projeto;
- Saída(s): Propósitos a serem atingidos pelas ferramentas.

Atividade 1.3 - Planejar Avaliação e Seleção de Ferramentas: corresponde à elaboração de um plano no qual devem ser tratados todos os aspectos relativos à gerência do processo de avaliação e seleção de ferramentas, tais como cronograma para execução das atividades, recursos humanos necessários e custo total, dentre outros itens que possam ser pertinentes;

- Entrada(s): Objetivos gerais do projeto e Propósitos a serem atingidos pelas ferramentas;

- Saída(s): Plano de projeto.

Atividade 1.4 - Estabelecer Requisitos Funcionais: tem como objetivo definir o conjunto de requisitos que devem ser atendidos pelas ferramentas, de forma a atingir os seus propósitos;

- Entrada(s): Propósitos a serem atingidos pelas ferramentas;
- Saída(s): Requisitos funcionais a serem atendidos pelas ferramentas.

Atividade 1.5 - Decompor Requisitos Funcionais: quando necessário, decompor os requisitos funcionais em sub-requisitos, de forma a detalhar o requisito funcional de maior nível;

- Entrada(s): Requisitos funcionais a serem atendidos pelas ferramentas;
- Saída(s): Requisitos e sub-requisitos funcionais a serem atendidos pelas ferramentas.

Atividade 1.6 - Priorizar Requisitos Funcionais: refere-se à classificação dos requisitos e, se necessário, à atribuição de pesos. A classificação tem o objetivo de estabelecer um conjunto de requisitos críticos, os quais. são requisitos considerados fundamentais para que uma ferramenta atinja um determinado propósito e, consequentemente, deveriam ser prioritariamente atendidos pelas ferramentas. Um exemplo é o requisito “suporte à criação, atualização e visualização da matriz de rastreabilidade”, que é fundamental para o propósito de apoiar a área de processo de Gerência de Requisitos. Já a atribuição de pesos tem o objetivo de, posteriormente, apoiar a seleção das ferramentas, ao permitir a obtenção de uma pontuação única referente aos requisitos funcionais. Para isso, pesos devem ser atribuídos às categorias de requisitos funcionais;

- Entrada(s): Requisitos e sub-requisitos funcionais a serem atendidos pelas ferramentas;
- Saída(s): Requisitos e sub-requisitos funcionais classificados e priorizados.

Atividade 1.7 - Estabelecer Requisitos Não Funcionais: tem como objetivo definir o conjunto de requisitos não funcionais que são relevantes de acordo com os propósitos estabelecidos;

- Entrada(s): Objetivos gerais do projeto e Propósitos a serem atingidos pelas ferramentas;
- Saída(s): Requisitos não funcionais identificados.

Atividade 1.8 - Decompor Requisitos Não Funcionais: quando necessário, decompor os requisitos não funcionais em sub-requisitos, de forma a detalhar o requisito não funcional de maior nível;

- Entrada(s): Requisitos não funcionais identificados;
- Saída(s): Requisitos e sub-requisitos não funcionais identificados.

Atividade 1.9 - Priorizar Requisitos Não Funcionais: refere-se à classificação e à atribuição de pesos a esses requisitos. A classificação tem o objetivo de organizar os requisitos, enquanto a atribuição de pesos tem o objetivo de permitir que uma pontuação referente aos requisitos não funcionais seja atribuída às ferramentas, considerando a relevância associada a cada requisito;

- Entrada(s): Requisitos e sub-requisitos não funcionais identificados;
- Saída(s): Requisitos e sub-requisitos não funcionais classificados e priorizados.

Atividade 1.10 - Verificar Consistência: analisar a consistência entre os requisitos estabelecidos para as ferramentas, assim como entre os requisitos das ferramentas e o objetivo do projeto. Caso estejam consistentes, passa-se para a macro-atividade de Avaliação. Caso contrário, é necessário revisar os requisitos até que se tornem consistentes;

- Entrada(s): Objetivos gerais do projeto, Propósitos a serem atingidos pelas ferramentas, Requisitos funcionais e Requisitos não funcionais;
- Saída(s): Os mesmos documentos, porém revisados e alterados, se necessário.

Macro-Atividade 2 - Avaliação

A proposta desta macro-atividade é identificar e avaliar um conjunto de potenciais ferramentas relacionadas aos propósitos estabelecidos. Cada ferramenta deve atender a um ou mais propósitos identificados na atividade 1.2

(Estabelecer Propósitos para as Ferramentas). Até que se obtenha um conjunto de ferramentas candidatas ou pelo menos uma boa ferramenta candidata (ISO/IEC, 1995), várias iterações dessa macro-atividade podem ocorrer. Um critério para dar continuidade ao processo de avaliação e seleção de ferramentas é o número de ferramentas identificadas. O ideal seria a identificação de aproximadamente 5 (cinco) ferramentas para cada propósito estabelecido.

As atividades de Avaliação, que foram adaptadas das normas ISO/IEC 14102 e ISO/IEC 14598, são apresentadas na Figura 7 – Atividades de Identificação, e detalhadas em seguida.

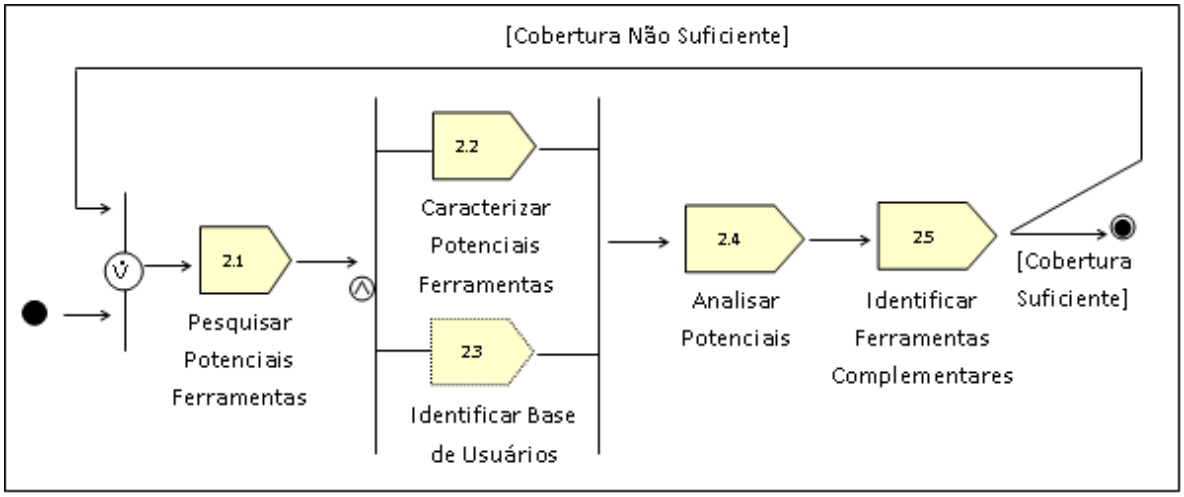


Figura 7 – Atividades de Identificação

Atividade 2.1 - Pesquisar Potenciais Ferramentas: refere-se à identificação de ferramentas CASE que possam ser úteis para os propósitos estabelecidos;

- Entrada(s): Propósitos a serem atingidos pelas ferramentas;
- Saída(s): Conjunto de ferramentas para cada um dos propósitos estabelecidos.

Atividade 2.2 - Caracterizar Potenciais Ferramentas: refere-se à obtenção e organização de informações sobre o conjunto de ferramentas identificadas. Algumas das informações de interesse são as características funcionais e não funcionais, o ambiente de execução e a infra-estrutura fornecida para manutenção e evolução da ferramenta;

- Entrada(s): Conjunto de ferramentas para cada um dos propósitos estabelecidos;
- Saída(s): Caracterização das potenciais ferramentas, incluindo nome da ferramenta, URL em que ela está hospedada, instituição ou pessoa mantenedora, objetivo declarado e propósito a ser atingido pela ferramenta, sistema(s) operacional(is), requisitos de hardware e software, e as principais características da ferramenta.

Atividade 2.3 - Identificar Base de Usuários: refere-se à obtenção de informações a respeito dos usuários que utilizam as ferramentas identificadas, as quais podem incluir tempo de utilização e opinião sobre pontos fortes e fracos. Esta atividade não é trivial, pois nem sempre há comunidade organizada, grupos de discussão sobre a ferramenta, uma lista de usuários disponível, ou responsável pela distribuição da mesma.No entanto, deve-se pelo menos tentar obter alguma informação em relação à base de usuários da ferramenta. De qualquer forma, a execução dessa atividade pode ser considerada opcional;

- Entrada(s): Caracterização das potenciais ferramentas;
- Saída(s): Caracterização das potenciais ferramentas, incluindo informações a respeito da base de usuários.

Atividade 2.4 - Analisar Potenciais Ferramentas: tem como objetivo avaliar as potenciais ferramentas e definir um conjunto de ferramentas candidatas à seleção. As ferramentas candidatas deveriam, de forma ideal, ter ampla base de usuários, o que significa que, provavelmente, já estariam bem consolidadas no mercado; e atender a um grande número de requisitos funcionais relacionados aos propósitos estabelecidos, especialmente de requisitos críticos, identificados na atividade 1.6 (Priorizar Requisitos Funcionais);

- Entrada(s): Caracterização das potenciais ferramentas e Requisitos e sub-requisitos funcionais classificados e priorizados;
- Saída(s): Potenciais ferramentas selecionadas para cada um dos propósitos estabelecidos.

Atividade 2.5 - Identificar Ferramentas Complementares: tem como objetivo identificar ferramentas que satisfaçam aos requisitos funcionais que não foram atendidos ou que foram atendidos fracamente na análise das potenciais ferramentas. A identificação de ferramentas complementares deve ser seguida da Atividade 2.2 - Caracterizar Potenciais Ferramentas e da Atividade 2.3 - Identificar Base de Usuários, como indicado na Figura 7;

- Entrada(s): Requisitos e sub-requisitos classificados e priorizados, e Potenciais ferramentas selecionadas para cada um dos propósitos estabelecidos;
- Saída(s): Caracterização das ferramentas complementares e Potenciais ferramentas complementares selecionadas para cada um dos propósitos estabelecidos.

Macro-Atividade 3 - Seleção

A proposta desta macro-atividade é escolher, dentre as ferramentas pré-selecionadas, a(s) ferramenta(s) CASE mais adequada(s) de acordo com os propósitos estabelecidos. Esta macro-atividade é composta das atividades apresentadas na Figura 8, as quais são detalhadas em seguida. Essas atividades foram adaptadas da norma ISO/IEC 14102.

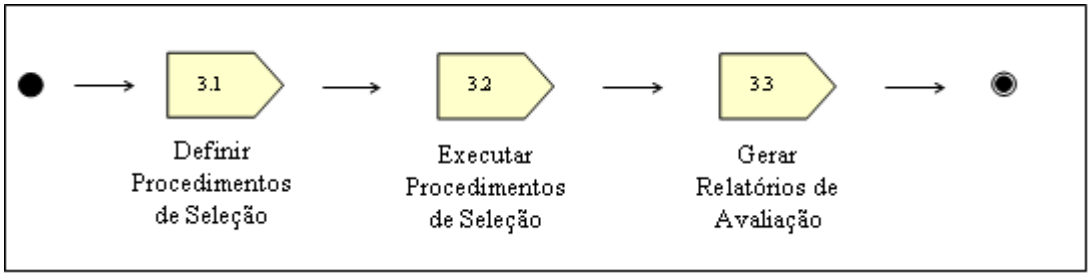


Figura 8 – Atividades de Seleção

Atividade 3.1 - Definir Procedimentos de Seleção: após identificar e analisar as ferramentas candidatas, é necessário definir um procedimento e critérios para a seleção da(s) melhor(es) ferramenta(s), considerando os requisitos definidos nas atividades 1.4 (Estabelecer Requisitos Funcionais) a 1.6 (Priorizar Requisitos

Funcionais) e 1.7 (Estabelecer Requisitos Não Funcionais) a 1.9 (Priorizar Requisitos Não Funcionais). É necessário também estabelecer escalas para determinação do grau em que os requisitos são atendidos pelas ferramentas, o que ajudará a atribuir pontuações para as ferramentas. O procedimento de seleção deve permitir selecionar a(s) ferramenta(s) que atenda(m) a um maior número de requisitos críticos, nas quais o atendimento a requisitos não funcionais também seja maximizado;

- Entrada(s): Objetivos gerais do projeto, Propósitos a serem atingidos pelas ferramentas, Requisitos funcionais e Requisitos não funcionais;
- Saída(s): Procedimento de seleção das ferramentas.

Atividade 3.2 - Executar Procedimento de Seleção: com o estabelecimento das escalas e a elaboração dos critérios de seleção, o procedimento de seleção deve ser executado para cada propósito estabelecido na Atividade 1.2 - Estabelecer Propósitos para as Ferramentas;

- Entrada(s): Procedimento de seleção das ferramentas, Requisitos funcionais, Requisitos não funcionais, Potenciais ferramentas selecionadas para cada um dos propósitos estabelecidos e Caracterização das potenciais ferramentas selecionadas, incluindo informações a respeito da base de usuários;
- Saída(s): Resultados do procedimento de seleção das ferramentas[1].

Atividade 3.3 - Gerar Relatório com Ferramentas Selecionadas: gerar um relatório contendo os dados relativos ao processo de avaliação e seleção das ferramentas, incluindo os objetivos gerais do projeto, os propósitos a serem atingidos pelas ferramentas, os requisitos funcionais e não funcionais identificados, os requisitos funcionais e não funcionais contemplados pela(s) ferramenta(s) selecionada(s) e em que nível estão contemplados.

- Entrada(s): Objetivos gerais do projeto, Propósitos a serem atingidos pelas ferramentas, Requisitos funcionais, Requisitos não funcionais e Resultados do procedimento de seleção das ferramentas;

[1] Tais documentos estarão especificados no procedimento de seleção

- Saída(s): Relatório com as ferramentas selecionadas, contendo informações sobre o processo de avaliação e seleção.

4.3 APLICAÇÃO DO PROCESSO

O processo descrito anteriormente foi aplicado para selecionar ferramentas livres que apóiam a obtenção dos resultados esperados no nível G do MPS.Br, composto pelos processos de Gerência de Requisitos e Gerência de Projetos. O resultado da execução do processo é apresentado a seguir.

Atividade 1.1 - Definir Objetivos do Projeto: como já mencionado, o objetivo do projeto foi selecionar ferramentas livres que fornecem suporte ferramental para a obtenção dos resultados esperados no nível G do MPS.Br.

Atividade 1.2 - Estabelecer Propósitos para as Ferramentas: os propósitos a serem atingidos pelas ferramentas foram os propósitos dos processos do nível G do MPS.Br, ou seja: 1) 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 necessário; e 2) gerenciar os requisitos do produto e dos componentes do produto, além de identificar inconsistências entre estes requisitos e os planos e produtos de trabalho do projeto.

Atividade 1.3 - Planejar Avaliação e Seleção de Ferramentas: a equipe de projeto foi formada por uma doutora, responsável pela orientação do trabalho, um mestrando, responsável pela condução da maioria das atividades, e por dois alunos de iniciação científica, responsáveis pela pesquisa e caracterização das ferramentas (Atividade 2.1 - Pesquisar Potenciais Ferramentas, e Atividade 2.2 - Caracterizar Potenciais Ferramentas, presentes na Figura 7 – Atividades de Identificação). Três computadores foram alocados ao projeto e três meses do cronograma foram destinados à seleção de ferramentas. Não foi elaborado um plano de custos, uma vez que o laboratório de Engenharia de Software da

Universidade Salvador (UNIFACS) foi cedido para o projeto e os envolvidos receberam bolsa de pesquisa para a realização do trabalho.

Atividade 1.4 - Estabelecer Requisitos Funcionais: os requisitos funcionais a serem atingidos pelas ferramentas foram estabelecidos com base nos resultados esperados dos processos do nível G do MPS.Br e nas práticas das áreas relacionadas do CMMI. Os requisitos funcionais de Gerência de Requisitos e de Gerência de Projeto encontram-se respectivamente descritos nos Quadros 5 e 6.

Atividade 1.5 - Decompor Requisitos Funcionais: os Quadros 5 e 6 mostram como os requisitos funcionais definidos para os propósitos de Gerência de Requisitos e Gerência de Projeto foram decompostos. Os requisitos aparecem nas linhas cinzas das tabelas e os respectivos sub-requisitos nas linhas brancas logo em seguida.

Atividade 1.6 - Priorizar Requisitos Funcionais: dentre os requisitos listados nos Quadros 5 e 6, foram identificados os requisitos críticos para cada um dos propósitos estabelecidos, os quais estão destacados em negrito e itálico. Tais requisitos críticos consideram as funcionalidades básicas que precisam ser oferecidas pelas ferramentas para atender ao propósito estabelecido. Como não faz sentido selecionar ferramentas que não satisfiçam ou satisfiçam pobremente os requisitos críticos, foi atribuído peso 5 aos requisitos críticos e peso 1 aos demais requisitos funcionais. Mais do que isso, foi decidido que ferramentas que não satisfizessem os requisitos críticos deveriam ser automaticamente descartadas

Requisitos/Sub-requisitos	**Peso**
[GR 1] **Gerência de Requisitos**: permitir o registro e a atualização dos requisitos técnicos e não técnicos do cliente, bem como dos estabelecidos pela organização desenvolvedora	
[GR 1.1] Registro e atualização de critérios para identificação dos fornecedores de requisitos apropriados	1
[GR 1.2] Registro e atualização de critérios para avaliação e aceitação de requisitos	1
[GR 1.3] ***Registro e atualização dos requisitos propriamente ditos, com identificação da fonte do requisito***	5
[GR 1.4] Registro da avaliação dos requisitos de acordo com os critérios estabelecidos	1
[GR 1.5] ***Manutenção do histórico de mudanças dos requisitos, incluindo justificativa***	5
[GR 1.6] Registro da avaliação dos produtos de trabalho e planos do projeto quanto à consistência com os requisitos, identificando fontes de inconsistências para correção	1

Requisitos/Sub-requisitos	Peso
[GR 2] **Gerência dos Compromissos com os Requisitos**: permitir o registro e a atualização dos compromissos assumidos pelos *stakeholders*	
[GR 2.1] Registro e atualização do compromisso com a especificação de requisitos	1
[GR 2.2] Registro e atualização do compromisso com uma mudança de requisito	1
[GR 3] **Gerência da Matriz de Rastreabilidade**: permitir o registro e a atualização da matriz de rastreabilidade, a qual registra as relações entre itens de decomposição do produto	
[GR 3.1] ***Registro e atualização da rastreabilidade horizontal, que captura as relações de dependência entre itens do mesmo nível de decomposição do produto***	5
[GR 3.2] Registro e atualização da rastreabilidade vertical, que captura as relações de origem entre itens que se encontram em diferentes níveis de decomposição do produto	1
[GR 3.3] Consulta à matriz de rastreabilidade para identificação dos itens aos quais um requisito está de alguma forma associado, visando determinação do possível impacto de uma mudança no requisito	1

Quadro 5 – Requisitos Funcionais para a Gerência de Requisitos

Requisitos/Sub-requisitos	Peso
[GP 1] **Gerência do Escopo do Projeto:** permitir o registro e a atualização do escopo do projeto	
[GP 1.1] Registro e atualização de uma WBS	1
[GP 1.2] Descrição dos pacotes de trabalho	1
[GP 1.3] Classificação dos pacotes de trabalho a serem desenvolvidos, adquiridos ou reutilizados	1
[GP 2] **Gerência do Processo da Execução do Projeto:** permitir o registro e a atualização das atividades do processo e da abordagem técnica a ser utilizada	
[GP 2.1] ***Registro e atualização das atividades do processo e suas dependências***	5
[GP 2.2] Registro e atualização da abordagem técnica adotada (DE, DOO, DOA, Arquitetura, LP)	1
[GP 3] **Gerência das Estimativas de Atributos das Tarefas e Produtos de Trabalho:** permitir o registro e a atualização das estimativas de tamanho, complexidade e recursos necessários	
[GP 3.1] ***Registro e atualização das estimativas de atributos das tarefas ou produtos de trabalho (Ex: tamanho e complexidade)***	5
[GP 3.2] Registro e a atualização das estimativas de recursos	1
[GP 3.3] Registro e a atualização do raciocínio dessas estimativas	1
[GP 4] **Definição do Ciclo de Vida do Projeto:** permitir o registro e a modificação, caso necessário, do ciclo de vida do projeto	
[GP 4.1] Registro e atualização do ciclo de vida a ser adotado no projeto	1
[GP 4.2] Registro e atualização do mapeamento entre o ciclo de vida adotado e atividades do processo	1
[GP 5] **Gerência das Estimativas de Esforço e Custo:** permitir o registro e a atualização das estimativas de esforço e custo	
[GP 5.1] ***Registro e atualização das estimativas de esforço***	5
[GP 5.2] ***Registro e atualização das estimativas de custo***	5
[GP 5.3] Registro e a atualização do raciocínio dessas estimativas	1
[GP 6] **Gerência do Cronograma e do Orçamento do Projeto:** permitir o registro e a atualização do cronograma e orçamento	
[GP 6.1] ***Registro e atualização do cronograma***	5
[GP 6.2] Registro e atualização do orçamento	1
[GP 6.3] Registro e a atualização do raciocínio utilizado na elaboração do cronograma e orçamento, incluindo premissas e restrições	1

Requisitos/Sub-requisitos	Peso
[GP 7] **Gerência de Riscos do Projeto:** permitir o registro e a atualização do plano de riscos do projeto, com a identificação do impacto e probabilidade dos mesmos bem como a definição das prioridades	1
[GP 8] **Gerência dos Dados do Projeto:** permitir o registro e a atualização dos artefatos a serem gerados ou utilizados no projeto, bem como de roteiros para elaboração dos artefatos	
[GP 8.1] Registro e atualização dos artefatos a serem gerados ou utilizados no projeto, identificando mídia, formato, público-alvo e forma de distribuição	1
[GP 8.2] ***Registro e atualização dos roteiros de artefatos, especificando conteúdo e formato dos mesmos***	5
[GP 9] **Gerência dos Recursos do Projeto:** permitir o registro e a atualização do plano de recursos do projeto, que podem ser humanos, de software, de hardware ou outros recursos	
[GP 9.1] Registro e atualização das competências requeridas no projeto	1
[GP 9.2] ***Registro e atualização do plano de recursos humanos, o qual aloca pessoas aos papéis previamente identificados***	5
[GP 9.3] Registro e atualização dos planos de capacitação ou contratação de pessoas	1
[GP 9.4] Registro e atualização dos planos de recursos em geral (software, hardware, infra-estrutura)	1
[GP 10] **Gerência do Envolvimento dos *Stakeholders*:** permitir o registro e a atualização dos *stakeholders* do projeto, especificando grau de interação e a responsabilidade dos mesmos, especialmente para os *stakeholders* que não fazem parte da equipe do projeto	1
[GP 11] **Consolidação do Plano do Projeto:** criar o documento do Plano de Projeto e possibilitar diferentes visões do mesmo documento	
[GP 11.1] Criação do documento do Plano de Projeto a partir das informações registradas ou dos documentos armazenados	1
[GP 11.2] Coexistência de diferentes visões do mesmo documento de acordo com o público-alvo	1
[GP 12] **Gerência dos Compromissos com o Plano do Projeto:** permitir o registro e a atualização dos compromissos assumidos pelos *stakeholders*	
[GP 12.1] Registro e atualização dos compromissos assumidos com o estabelecido no Plano do Projeto	1
[GP 12.2] Indicação da necessidade de obter novo comprometimento com o Plano do Projeto	1
[GP 13] **Monitoração dos Parâmetros do Projeto:** permitir o registro de dados reais e/ou desvios com a possibilidade de registro da necessidade de uma ação corretiva referente aos parâmetros do projeto	
[GP 13.1] ***Registro de dados reais, desvios e necessidade de ação corretiva referentes ao tamanho e complexidade dos pacotes de trabalho e/ou tarefas***	5
[GP 13.2] ***Registro de dados reais, desvios e necessidade de ação corretiva referentes a esforço***	5
[GP 13.3] ***Registro de dados reais, desvios e necessidade de ação corretiva referentes a cronograma***	5
[GP 13.4] ***Registro de dados reais, desvios e necessidade de ação corretiva referentes a custos***	5
[GP 13.5] ***Registro de dados reais, desvios e necessidade de ação corretiva referentes a recursos***	5
[GP 13.6] Registro de dados reais, desvios e necessidade de ação corretiva referentes ao desenvolvimento das competências requeridas no projeto	1
[GP 14] **Monitoração dos Compromissos:** permitir a identificação dos compromissos que não estão sendo satisfeitos ou que estão em risco de não serem satisfeitos com a possibilidade de registro da necessidade de uma ação corretiva	1
[GP 15] **Monitoração dos Riscos do Projeto:** permitir o registro de ocorrência de um risco, a indicação de exclusão/inclusão de riscos na lista de riscos, bem como indicação de alterações no impacto e probabilidade dos mesmos	1

Requisitos/Sub-requisitos	Peso
[GP 16] **Monitoração da Gerência dos Dados do Projeto:** permitir o registro de desvios na geração ou utilização de artefatos com a possibilidade de registro da necessidade de uma ação corretiva	1
[GP 17] **Monitoração do Envolvimento dos *Stakeholders*:** permitir o registro de desvios nos envolvimentos previstos de *stakeholders* com a possibilidade de registro da necessidade de uma ação corretiva	1
[GP 18] **Condução de Reuniões em Pontos e Marcos de Controle:** permitir o registro de reuniões ocorridas com o intuito de monitorar o projeto (data, hora, participantes, problemas identificados, solicitações, decisões e itens de ação) em complementação aos desvios já tratados nos requisitos anteriores	1

Quadro 6 – Requisitos Funcionais para a Gerência de Projeto

Atividade 1.7 - Estabelecer Requisitos Não Funcionais: os requisitos não funcionais foram obtidos das normas ISO/IEC 14102 (*Tecnologia de Informação - Guia para Avaliação e Seleção*) e ISO/IEC 9126-1 (*Engenharia de Software - Qualidade de Produto. Parte 1 - Modelo de Qualidade*). De acordo com o proposto nas normas, foram obtidas as seguintes características: Funcionalidade, Interoperabilidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade. A característica Interoperabilidade é uma sub-característica de Funcionalidade na norma ISO/IEC 9126-1, porém na norma ISO/IEC 14102 ela aparece como uma característica, contendo outras sub-características. Por ser fundamental para o objetivo mais amplo de prover uma infra-estrutura de integração para as ferramentas selecionadas, a interoperabilidade foi considerada como uma característica, o que permitiu avaliar a importância de suas sub-características. Os requisitos não funcionais, derivados das características acima, encontram-se listados no Quadro 7 e são válidos para ambos os propósitos estabelecidos.

Atividade 1.8 - Decompor Requisitos Não Funcionais: os requisitos não funcionais foram decompostos também com base nas normas ISO/IEC 14102 e ISO/IEC 9126-1. No entanto, a sub-característica *Freqüência de Atualizações*, que não é citada nas normas consultadas, foi incluída em *Manutenibilidade*, por ter sido considerada importante no contexto de ferramentas livres. Os requisitos aparecem nas linhas cinzas do Quadro 7 e os respectivos sub-requisitos nas linhas brancas logo em seguida.

Atividade 1.9 - Priorizar Requisitos Não Funcionais: nesta atividade, um questionário foi aplicado com 12 pessoas envolvidas em esforços de melhoria de processos de software. Deste total, 6 pessoas tiveram participação direta em projetos bem sucedidos de implantação do CMMI. Para a resposta dessas pessoas foi adotado peso 2, enquanto para a resposta das demais foi adotado peso 1. No questionário, elas tiveram que atribuir, para cada um dos potenciais requisitos não funcionais, uma das seguintes pontuações: 0 (Sem Importância), 1 (Pouca Importância), 2 (Importante), 3 (Muito Importante) e 4 (Imprescindível). O resultado da aplicação do questionário, levando em consideração o peso dos avaliadores e a resposta dada por eles, foi tabulado, resultando em um peso para cada um dos requisitos não funcionais. O questionário com a tabulação dos resultados está disponível no Apêndice A. Os pesos obtidos são apresentados no Quadro 7, sendo válidos para ambos os propósitos estabelecidos.

Requisitos/Sub-requisitos		**Peso**
Funcionalidade: existência de um conjunto de funções que atendam às necessidades explícitas ou implícitas do usuário, quando a ferramenta estiver sendo utilizada sob as condições especificadas		
Adequação	Capacidade da ferramenta de prover um conjunto apropriado de funções para os objetivos e tarefas que foram especificados pelo usuário	3,3
Acurácia (Exatidão)	Capacidade da ferramenta de prover resultados ou efeitos corretos ou conforme acordados	3,8
Segurança de Acesso	Capacidade da ferramenta de evitar o acesso não autorizado a informações e dados	2,8
Interoperabilidade: capacidade da ferramenta de interagir com uma ou mais ferramentas especificadas		
Integração de Dados	Capacidade da ferramenta de usar, processar e distribuir informações compartilhadas com um conjunto de ferramentas	3,2
Integração de Interface	Capacidade da ferramenta de apresentar uma interface consistente com um conjunto de ferramentas	1,8
Integração de Controle	Capacidade da ferramenta de interagir com um conjunto de ferramentas ou ambientes de software	2,3
Integração de Conhecimento	Capacidade da ferramenta de disponibilizar o acesso aos seus metadados	2,3
Confiabilidade: capacidade da ferramenta de manter um nível de desempenho especificado, quando usada sob as condições especificadas		
Maturidade	Capacidade da ferramenta de evitar falhas decorrentes de defeitos na mesma	3,2
Tolerância a Falhas	Capacidade da ferramenta de manter um nível de desempenho especificado em casos de falha da ferramenta ou de violação da interface especificada	2,6
Recuperabilidade	Capacidade da ferramenta de re-estabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha	3,1
Utilização de Rotinas de Backup	Capacidade da ferramenta de prover e utilizar backup	2,9

Requisitos/Sub-requisitos		Peso
Usabilidade: capacidade da ferramenta de ser compreendida, aprendida e operada de forma atraente ao usuário, quando usada sob as condições especificadas		
Inteligibilidade	Capacidade da ferramenta de possibilitar ao usuário compreender que ela é apropriada para as tarefas e condições de uso especificadas	3,1
Apreensibilidade	Capacidade da ferramenta de possibilitar ao usuário aprender a sua utilização	2,7
Operacionalidade	Capacidade da ferramenta de permitir que o usuário a opere facilmente	2,9
Atratividade	Capacidade da ferramenta de ser atraente ao usuário	1,9
Eficiência: capacidade da ferramenta de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob as condições especificadas		
Comportamento em relação ao Tempo	Capacidade da ferramenta de oferecer taxa de transferência e tempo de resposta e de processamento apropriados, quando executada sob as condições especificadas	3,1
Utilização de Recursos	Capacidade da ferramenta de usar o tipo e quantidade de recursos apropriados, quando executada sob as condições especificadas	2,7
Manutenibilidade: capacidade da ferramenta de ser modificada. As modificações podem incluir correções, melhorias ou adaptações		
Analisabilidade	Capacidade da ferramenta de permitir o diagnóstico de defeitos ou causas de falhas, e/ou identificação de partes a serem modificadas	2,7
Modificabilidade	Capacidade da ferramenta de permitir que uma modificação especificada seja implementada	3,1
Estabilidade	Capacidade da ferramenta de evitar efeitos inesperados decorrentes de modificações	3,1
Testabilidade	Capacidade da ferramenta de permitir que ela seja validada, quando modificada	2,9
Frequência de Atualizações	Política da organização de fornecer atualizações regulares para corrigir problemas ou incluir funcionalidades	2,7
Portabilidade: capacidade da ferramenta de permitir sua transferência de um ambiente para outro		
Adaptabilidade	Capacidade da ferramenta de ser adaptada para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pela ferramenta	2,4
Capacidade para ser instalado	Capacidade da ferramenta de ser instalada em um ambiente especificado	3,2
Coexistência	Capacidade da ferramenta de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos	3,4
Capacidade para substituir	Capacidade da ferramenta de ser usada em substituição a outra ferramenta de software especificada, com o mesmo propósito e no mesmo ambiente	2,4

Quadro 7 – Requisitos Não Funcionais para todas Áreas de Processo

Atividade 1.10 - Verificar Consistência: todos os requisitos identificados foram verificados de acordo com os propósitos estabelecidos. Além disso, a consistência entre os requisitos não funcionais foi verificada, assim como a consistência entre os requisitos não funcionais e o objetivo do projeto.

Atividade 2.1 - Pesquisar Potenciais Ferramentas: foram pesquisadas ferramentas em repositórios de ferramentas livres, como www.sourceforge.net, www.tigris.org e www.savannah.nongnu.org/, além de sites de busca. Todas as ferramentas livres encontradas que se propunham a gerenciar requisitos e gerenciar projetos de software foram consideradas. Para gerência de requisitos, foram encontradas 33 ferramentas, enquanto que para gerência de projeto foram encontradas 34 ferramentas. As ferramentas pesquisadas encontram-se listadas no Apêndice B.

Atividade 2.2 - Caracterizar Potenciais Ferramentas: todas as ferramentas foram caracterizadas segundo modelo de documento presente no Apêndice C.

Atividade 2.3 - Identificar Base de Usuários: apesar de estar previsto no processo, não foi possível obter muitas informações a respeito da base de usuário das ferramentas encontradas, devido à falta de informações nas páginas de onde as ferramentas foram obtidas.

Atividade 2.4 - Analisar Potenciais Ferramentas: as ferramentas encontradas foram analisadas segundo alguns critérios, como real possibilidade de ser utilizada (algumas ferramentas ainda não estavam disponíveis para download), estabilidade (quando instaladas e executadas, algumas ferramentas se mostraram muito instáveis) e atendimento aos requisitos funcionais críticos. O resultado da análise quanto ao atendimento aos requisitos funcionais críticos é apresentado no Quadro 8, no que se refere à Gerência de Requisitos, e no Quadro 9, no que se refere à Gerência de Projeto. Para a Gerência de Requisitos, apenas 3 ferramentas foram pré-selecionadas: RTH, OSRMT e Controla, sendo descartada a CRM-Ctt. Para a Gerência de Projeto, foram pré-selecionadas a DotProject, NetOffice, GP Tool, Open WorkBench, PHProjekt e Streber, sendo descartadas a Project Planner, a GanttPV e a JProject Timer.

REQUISITOS CRÌTICOS	FERRAMENTAS			
	Controla	CRM-Ctt	OSRMT	RTH
[GR 1.3]	Sim	Sim	Sim	Sim
[GR 1.5]	Sim	Sim	Sim	Sim
[GR 3.3]	Sim	Não	Sim	Sim

Quadro 8 – Avaliação das Ferramentas para Gerência de Requisitos segundo os Requisitos Críticos

REQUISITOS CRÌTICOS	FERRAMENTAS								
	Dotproject	NetOffice	Open WorkBench	Project Planner	GanttPV	JProject Timer	GP Tool	PHProjekt	Streber
[GP 2.1]	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
[GP 3.1]	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
[GP 5.1]	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
[GP 5.2]	Sim	Sim	Sim	Sim	Não	Não	Sim	Sim	Sim
[GP 6.1]	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim	Sim
[GP 8.2]	Sim	Sim	Sim	Não	Não	Não	Sim	Sim	Sim
[GP 9.2]	Sim	Sim	Sim	Sim	Sim	Não	Sim	Sim	Sim
[GP 13.1]	Sim	Sim	Sim	Sim	Não	Não	Sim	Sim	Sim
[GP 13.2]	Sim	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim
[GP 13.3]	Sim	Sim	Sim	Sim	Não	Não	Sim	Sim	Sim
[GP 13.4]	Sim	Sim	Sim	Sim	Não	Não	Sim	Sim	Sim
[GP 13.5]	Sim	Sim	Sim	Sim	Não	Não	Sim	Sim	Sim

Quadro 9 – Avaliação das Ferramentas para Gerência de Requisitos segundo os Requisitos Críticos

Atividade 2.5 - Identificar Ferramentas Complementares: esta atividade não foi executada, ou seja, não foram identificadas ferramentas complementares às pré-selecionadas na atividade anterior. Isto porque as ferramentas pré-selecionadas satisfizeram os requisitos funcionais críticos para cada um dos propósitos estabelecidos e uma cobertura completa dos requisitos funcionais estaria além do escopo desta dissertação.

Atividade 3.1 - Definir Procedimentos de Seleção: para a avaliação dos requisitos funcionais, foi definida a seguinte escala: *0 – Não atende ao requisito*, atribuído quando a ferramenta não permite nem mesmo a inclusão de um arquivo contendo as informações mencionadas no requisito avaliado; *1 – Armazenamento de Documento contendo Informações*, atribuído quando a ferramenta apenas permite a inclusão de um arquivo contendo as informações mencionadas no requisito avaliado; *2 – Registro e Atualização de Informações*, quando a ferramenta permite incluir as informações relacionadas ao requisito de forma estruturada, bem como permite a sua atualização; *3 – Apoio Básico à Geração das Informações (através de um método/técnica ou auxílio de uma base de conhecimento)*, atribuído quando a ferramenta disponibiliza uma base de conhecimento ou propõe/apóia o uso de algum método/técnica para a aquisição das informações referentes ao requisito (ex.: apoio ao uso da técnica de pontos por função para obter tamanho de um módulo do projeto); *e 4 – Apoio Avançado à Geração das Informações (através de mais de um método/técnica, ou da combinação de um método/técnica com o auxílio fornecido por uma base de conhecimento)*. Esta última é semelhante à anterior, porém permite a comparação de informações obtidas por diferentes meios. O peso atribuído a cada requisito e valor obtido com base na escala elaborada geram uma pontuação para as ferramentas referentes aos requisitos funcionais, a qual serve como base para a seleção das ferramentas.

Para a avaliação de cada um dos requisitos não funcionais, foi selecionada uma métrica da norma ISO/IEC 9126-2 (*Engenharia de Software - Qualidade de Produto. Parte 2 – Métricas Externas*). Para cada métrica, foi utilizada a fórmula exatamente como proposta pela norma ou, então, com pequenas adaptações. Os

requisitos não funcionais de Integração de Interface, Integração de Controle, Integração de Conhecimento, Utilização de Rotinas de Backup e Freqüência de Atualizações não foram obtidos da norma ISO/IEC 9126-1 e, conseqüentemente, não puderam ter suas métricas obtidas diretamente da norma ISO/IEC 9126-2. Neste caso, as métricas foram definidas com base em métricas similares encontradas na ISO/IEC 9126-2. Para cada métrica, o domínio de valores possíveis estabelecido pela fórmula foi dividido em três sub-intervalos, os quais foram mapeados para a seguinte escala: *0 – Não Atende ao Requisito*, *1 – Atende Parcialmente ao Requisito* e *2 – Atende Totalmente ao Requisito*. Para exemplificar: a métrica "Adequação Funcional" foi adotada para apoiar a avaliação do requisito não funcional Adequação e sua fórmula é "X=1-A/B, onde A é o número de funções nas quais problemas são detectados na avaliação e B é o número total de funções avaliadas". Neste caso, quanto mais perto de 1 o resultado da medição for, melhor o requisito é atendido. Logo, os sub-intervalos adotados foram: de 0 a 0,4 (0 – *Não Atende ao Requisito*); de 0,5 a 0,7 (1 – *Atende Parcialmente ao Requisito*); e de 0,8 a 1 (2 - *Atende Totalmente ao Requisito*). Para os requisitos não funcionais com avaliação binária, ou seja, que estão ou não estão presentes na ferramenta, foi utilizada uma escala mais simples, apenas com os valores: 0 – *Não Atende ao Requisito* e 2 – *Atende Totalmente ao Requisito.* Utilização de Rotinas de Backup é um exemplo de tais requisitos. A relação das métricas utilizadas para cada requisito não funcional é apresentada no Apêndice D.

Para a seleção da(s) melhor(es) ferramenta(s) livre(s) para os respectivos propósitos, foi definido que as pontuações referentes aos requisitos funcionais e não funcionais seriam consideradas em conjunto, ou seja, ferramentas que atendessem mais adequadamente a um maior conjunto de requisitos funcionais e não funcionais teriam prioridade na seleção. Em caso de ferramentas com pontuações semelhantes (empate técnico), o atendimento aos requisitos funcionais críticos deveriam ser utilizados para concretizar a seleção.

Atividade 3.2 - Executar Procedimento de Seleção: o procedimento de seleção foi executado para as ferramentas pré-selecionadas. Nenhuma das ferramentas para os processos de Gerência de Requisitos e Gerência de Projetos obtiveram a pontuação 3 (*Apoio Básico à Geração das Informações*) ou 4 (*Apoio Avançado à Geração das Informações*) na avaliação de qualquer dos requisitos funcionais. A Tabela 1 apresenta o resultado da avaliação dos requisitos funcionais para o propósito de Gerência de Requisitos, enquanto que a, Tabela 2, para o propósito de Gerência de Projetos. O resultado da avaliação dos requisitos não funcionais para cada propósito é apresenta, respectivamente, nas Tabelas 3 e 4. A descrição completa da avaliação dos requisitos não funcionais é fornecida no Apêndice E.

Tabela 1 – Avaliação de Requisitos Funcionais para Gerência de Requisitos

Requisitos Decompostos	**Peso**	**Ferramentas**		
		RTH	**Controla**	***OSRMT***
[GR 1.1]	1	2	0	1
[GR 1.2]	1	2	2	1
[GR 1.3]	5	2	2	2
[GR 1.4]	1	2	2	1
[GR 1.5]	5	2	2	2
[GR 1.6]	1	2	2	1
[GR 2.1]	1	2	2	2
[GR 2.2]	1	2	2	2
[GR 3.1]	1	2	2	2
[GR 3.2]	1	2	2	2
[GR 3.3]	5	2	2	2
TOTAL		46	44	42

Tabela 2 – Avaliação de Requisitos Funcionais para Gerência de Projetos

Requisitos Decompostos	Peso	Ferramentas					
		DotProject	NetOffice	GP Tool	Open WorkBench	PHProjekt	Streber
[GP 1.1]	1	2	2	2	2	2	2
[GP 1.2]	1	2	2	2	2	2	2
[GP 1.3]	1	2	2	0	2	1	2
[GP 2.1]	5	2	2	2	2	2	2
[GP 2.2]	1	2	2	0	2	1	2
[GP 3.1]	5	2	2	2	2	2	2
[GP 3.2]	1	2	1	2	2	1	0
[GP 3.3]	1	2	2	2	2	1	0
[GP 4.1]	1	2	2	0	2	2	2
[GP 4.2]	1	2	2	0	2	1	2
[GP 5.1]	5	2	2	2	2	2	2
[GP 5.2]	5	2	1	2	2	2	2
[GP 5.3]	1	2	1	0	2	2	1
[GP 6.1]	5	2	2	2	2	2	2
[GP 6.2]	1	2	2	0	2	2	1
[GP 6.3]	1	2	2	2	2	1	1
[GP 7]	1	2	1	0	0	1	1
[GP 8.1]	1	1	1	0	0	1	2
[GP 8.2]	1	1	1	0	0	1	2
[GP 9.1]	1	2	2	1	1	1	1
[GP 9.2]	1	2	2	1	1	1	1
[GP 9.3]	1	1	1	0	1	1	1
[GP 9.4]	1	2	1	2	2	1	1
[GP 10]	1	2	2	2	2	2	2
[GP 11.1]	1	0	2	0	0	0	0
[GP 11.2]	1	0	2	0	0	0	0
[GP 12.1]	1	2	2	2	2	2	2
[GP 12.2]	1	2	2	2	2	2	2
[GP 13.1]	5	2	2	2	2	2	2
[GP 13.2]	5	2	2	2	2	2	2
[GP 13.3]	5	2	2	2	2	2	2
[GP 13.4]	5	2	2	2	2	2	2
[GP 13.5]	5	2	2	2	2	1	1
[GP 13.6]	1	2	2	2	2	1	1
[GP 14]	1	1	1	0	1	1	1
[GP 15]	1	2	2	0	0	0	1
[GP 16]	1	1	2	0	0	1	2
[GP 17]	1	1	2	0	2	2	2
[GP 18]	1	2	0	2	0	2	2
TOTAL		160	155	143	157	144	151

Tabela 3 – Avaliação de Requisitos Não Funcionais para Gerência de Requisitos (Continua)

Funcionalidade				
		RESULTADOS		
Sub-Característica	**PESO**	**RTH**	**Controla**	**OSRMT**
Adequação	3,3	2	2	2
Acurácia (Exatidão)	3,8	2	2	2
Segurança de Acesso	2,8	2	1	2
TOTAL		19,8	17	19,8
Interoperabilidade				
		RESULTADOS		
Sub-Característica	**PESO**	**RTH**	**Controla**	**OSRMT**
Integração de Dados	3,2	2	2	0
Integração de Interface	1,8	0	0	0
Integração de Controle	2,3	0	0	0
Integração de Conhecimento	2,3	2	2	2
TOTAL		11	11	4,6
Confiabilidade				
		RESULTADOS		
Sub-Característica	**PESO**	**RTH**	**Controla**	**OSRMT**
Maturidade	3,2	1	1	1
Tolerância a Falhas	2,6	2	2	1
Recuperabilidade	3,1	0	0	0
Utilização de Rotinas de Backup	2,9	0	0	0
TOTAL		8,4	8,4	5,8
Usabilidade				
		RESULTADOS		
Sub-Característica	**PESO**	**RTH**	**Controla**	**OSRMT**
Inteligibilidade	2,1	2	2	2
Apreensibilidade	2,7	1	2	0
Operacionalidade	2,9	1	2	1
Atratividade	1,9	1	2	1
TOTAL		11,7	19,2	9
Eficiência				
		RESULTADOS		
Sub-Característica	**PESO**	**RTH**	**Controla**	**OSRMT**
Comportamento em relação ao Tempo	3,1	2	2	2
Utilização de Recursos	2,7	2	2	0
TOTAL		11,6	11,6	6,2
Manutenibilidade				
		RESULTADOS		
Sub-Característica	**PESO**	**RTH**	**Controla**	**OSRMT**
Analisabilidade	2,7	2	2	2
Modificabilidade	3,1	1	1	2
Estabilidade	3,1	2	2	2
Testabilidade	2,9	2	2	2
Frequência de Atualizações	2,7	2	2	2
TOTAL		25,9	25,9	29

Tabela 3 – Avaliação de Requisitos Não Funcionais para Gerência de Requisitos (Conclusão)

Portabilidade				
		RESULTADOS		
Sub-Característica	**PESO**	**RTH**	**Controla**	**OSRMT**
Adaptabilidade	2,4	2	2	2
Capacidade para ser instalado	3,2	1	1	1
Coexistência	3,4	2	2	2
Capacidade para substituir	2,4	2	1	2
TOTAL		19,6	17,2	19,6
	TOTAL GERAL	**108**	**110,3**	**94**

Tabela 4 – Avaliação de Requisitos Não Funcionais para Gerência de Projetos (Continua)

Funcionalidade							
		RESULTADOS					
Sub-Carac.	**PESO**	**DotProject**	**NetOffice**	**GPTool**	**Open WorkBench**	**PHProjekt**	**Streber**
Adequação	3,3	2	2	2	2	1	1
Acurácia (Exatidão)	3,8	2	2	1	2	2	2
Segurança de Acesso	2,8	2	2	0	0	1	2
TOTAL		19,8	19,8	10,4	14,2	13,7	16,5
Interoperabilidade							
		RESULTADOS					
Sub-Carac.	**PESO**	**DotProject**	**NetOffice**	**GPTool**	**Open WorkBench**	**PHProjekt**	**Streber**
Integração Dados	3,2	2	2	1	0	1	0
Integração Interface	1,8	2	2	0	1	1	2
Integração Controle	2,3	2	0	0	0	0	0
Integração Conhecimento	2,3	2	2	1	2	1	2
TOTAL		19,2	14,6	5,5	6,4	7,3	8,2
Confiabilidade							
		RESULTADOS					
Sub-Carac.	**PESO**	**DotProject**	**NetOffice**	**GPTool**	**Open WorkBench**	**PHProjekt**	**Streber**
Maturidade	3,2	1	2	1	2	1	2
Tolerância a Falhas	2,6	2	2	2	2	2	2
Recuperabili-dade	3,1	0	0	0	0	0	0
Rotina Backup	2,9	2	2	0	0	0	0
TOTAL		14,2	17,4	8,4	11,6	8,4	11,6
Usabilidade							
		RESULTADOS					
Sub-Carac.	**PESO**	**DotProject**	**NetOffice**	**GPTool**	**Open WorkBench**	**PHProjekt**	**Streber**
Inteligibilidade	2,1	2	2	2	2	1	2
Apreensibilidade	2,7	2	2	1	2	1	2
Operacionalidade	2,9	1	2	0	0	1	0
Atratividade	1,9	1	2	1	1	0	2
TOTAL		14,4	19,2	8,8	11,5	7,7	13,4

Tabela 4 – Avaliação de Requisitos Não Funcionais para Gerência de Projetos (Conclusão)

Eficiência							
		RESULTADOS					
Sub-Carac.	**PESO**	**DotProject**	**NetOffice**	**GPTool**	**Open WorkBench**	**PHProjekt**	**Streber**
Comp. relação Tempo	3,1	2	2	2	2	2	2
Uso de Recurso	2,7	2	2	1	0	1	2
TOTAL		11,6	11,6	8,9	6,2	8,9	11,6
Manutenibilidade							
		RESULTADOS					
Sub-Carac.	**PESO**	**DotProject**	**NetOffice**	**GPTool**	**Open WorkBench**	**PHProjekt**	**Streber**
Analisablidade	2,7	2	2	2	2	2	2
Modificabilidade	3,1	2	1	0	0	0	0
Estabilidade	3,1	2	2	2	2	2	2
Testabilidade	2,9	2	2	2	2	2	2
Freq Atualização	2,7	2	2	2	2	2	2
TOTAL		29	25,9	22,8	22,8	22,8	22,8
Portabilidade							
		RESULTADOS					
Sub-Carac.	**PESO**	**DotProject**	**NetOffice**	**GPTool**	**Open WorkBench**	**PHProjekt**	**Streber**
Adaptabilidade	2,4	2	2	2	2	2	2
Capac. Instalação	3,2	2	2	2	2	1	2
Coexistência	3,4	2	2	2	2	2	2
Capacidade Subst.	2,4	2	2	1	2	2	2
TOTAL		22,8	22,8	20,4	22,8	19,6	22,8
TOTAL GERAL		**131**	**131,3**	**85,2**	**95,5**	**88,4**	**106,9**

Atividade 3.3 - Gerar Relatório com Ferramentas Selecionadas: baseado na caracterização das ferramentas e na pontuação dos requisitos funcionais e não funcionais, foram gerados relatórios contendo a avaliação das ferramentas. Para o propósito de Gerência de Requisitos, a ferramenta ***Controla*** obteve a maior pontuação nos requisitos não funcionais, porém possui alguns pontos negativos que influenciaram na sua não seleção, os quais são: funciona exclusivamente em ambiente Windows, utiliza um banco de dados proprietário (Microsoft Access), e não permite personalizar sua instalação (ex.: escolher diretório de instalação). A ferramenta ***OSRMT***, diferente da Controla, funciona em diversas plataformas e também atendeu bem aos requisitos funcionais. No entanto, se mostrou instável durante a execução, gerando falhas em funcionalidades críticas, como a inclusão de requisitos, o que prejudicou na pontuação final. A ferramenta ***RTH*** foi a escolhida porque: 1) foi a que melhor atendeu aos requisitos funcionais; 2) foi a que se apresentou mais estável durante a execução, apesar de ter apresentado

falhas ocasionais em algumas funcionalidades não críticas (ex.: envio de e-mail quando da alteração do estado de um requisito), as quais não chegaram a prejudicar as demais funcionalidades ou causar grande incômodo; 3) é a única que permite acesso remoto por ser baseada em ambiente web, o que facilita a sua utilização em ambientes geograficamente dispersos; e 4) é independente de sistema operacional, podendo ser migrada para qualquer plataforma sem necessidades de maiores esforços para a adaptação. O resultado da avaliação dos requisitos funcionais e não funcionais das ferramentas de Gerência de Requisitos é fornecido na Tabela 5.

Tabela 5 – Relatório de Avaliação para Gerência de Requisitos

FERRAMENTAS	REQUISITOS		TOTAL GERAL
	Funcionais	Não Funcionais	
RTH	46	110	**156**
Controla	44	112,3	**156,3**
OSRMT	42	96	**138**

Já para o propósito de Gerência de Projetos, as ferramentas ***GP Tool*** e ***PHP Projekt*** não obtiveram boa pontuação na avaliação dos requisitos funcionais e não funcionais. A ***GP Tool***, apesar de atender a todos os requisitos críticos de Gerência de Projetos, não atendeu bem aos demais requisitos funcionais, além de não ter estabilidade na execução e apresentar desvantagens em relação às concorrentes, como, por exemplo, a não inclusão de arquivos relacionados aos projetos e a não inclusão de observações relacionadas aos projetos e tarefas. A ***PHP Projekt*** atendeu bem aos requisitos críticos, porém também não atendeu bem aos demais requisitos funcionais, bem como foi a ferramenta que apresentou o pior resultado na avaliação dos requisitos não funcionais, devido à instabilidade em algumas funcionalidades e usabilidade não tão boa quanto a das ferramentas selecionadas. A ferramenta ***Open WorkBench*** também atendeu bem aos requisitos críticos de Gerência de Projetos, porém também não permite a inclusão de arquivos relacionados às funcionalidades fornecidas. Outra desvantagem é que a ***Open WorkBench*** é uma aplicação cliente-servidor, o que traz dificuldades em caso de projetos dispersos geograficamente. Finalmente, a ferramenta ***Streber***,

que também atendeu relativamente bem aos requisitos e possui certa estabilidade, porém as ferramentas selecionadas (dotProject e NetOffice) obtiveram melhor pontuação na avaliação dos requisitos funcionais e não funcionais, além de oferecerem outras funcionalidades, ou seja, são mais completas.

As duas ferramentas que melhor atenderam ao propósito de Gerência de Projetos foram a ***DotProject*** e ***NetOffice***. Ambas atenderam bem os requisitos funcionais, além de apresentar certa estabilidade durante a execução. A ferramenta ***NetOffice*** possibilita a geração de relatórios para o cliente, além de uma funcionalidade chamada "Site do Projeto", na qual tarefas e arquivos podem ser disponibilizados para clientes. No entanto, a ferramenta ***DotProject*** é mais completa, pois mostra melhor a estrutura de alto nível do projeto (WBS), incluindo tarefas filhas; possibilita associar custos para o projeto e inclusão de recursos em geral (hardware, software, etc...); contém plug-ins que podem ser instalados e possibilitam adicionar novas funcionalidades, como, por exemplo, o módulo de riscos de projeto. A desvantagem do ***DotProject*** é que ele não permite customizar a ferramenta para utilização pelos clientes e não tem geração de relatórios como a anterior. O resultado da avaliação dos requisitos funcionais e não funcionais das ferramentas de Gerência de Projetos é fornecido na Tabela 6.

Tabela 6 – Relatório de Avaliação para Gerência de Projeto

FERRAMENTAS	**REQUISITOS**		**TOTAL GERAL**
	Funcionais	**Não Funcionais**	
DotProject	160	131	**291**
NetOffice	155	131,3	**286,3**
GP Tool	143	85,2	**228,2**
Open WorkBench	157	95,5	**252,5**
PHP Projekt	144	88,4	**232,4**
Streber	151	106,9	**257,9**

4.4 CONCLUSÕES

Neste capítulo foi apresentado um processo para a avaliação e seleção de ferramentas livres, no qual foram definidos os requisitos para atender aos propósitos de Gerência de Requisitos e Gerência de Projetos. Para o primeiro propósito, a ferramenta selecionada foi a RTH, disponível em http://www.rth-is-quality.com/home.php, enquanto que para o segundo propósito, a ferramenta mais adequada foi a dotProject, disponível em http://www.dotproject.net.

No próximo capitulo, será utilizado o método para elaboração de ontologias proposto por Falbo (1998), tendo sido a primeira etapa executada conforme proposto por Grüninger e Fox (1995). O objetivo principal das ontologias é apoiar a integração dessas ferramentas.

CAPÍTULO 5. ONTOLOGIAS PARA APOIO À GERÊNCIA DE REQUISITOS E DE PROJETO

Neste capítulo, são apresentadas duas ontologias, a de Gerência de Requisitos e a de Gerência de Projetos, as quais foram elaboradas com dupla finalidade: auxiliar na identificação dos requisitos de informação de ferramentas que pretendem apoiar essas áreas de processo e permitir a integração de tais ferramentas, ao atuar como interlíngua.

5.1 VISÃO GERAL

Segundo Falbo (1998), a integração de ferramentas é normalmente tratada em três dimensões: apresentação, controle e dados. A *Integração de Apresentação* "é responsável por proporcionar ao usuário a sensação de integração do ambiente" formado pelo conjunto de ferramentas (FALBO, 1998, p. 35), sendo percebida tanto pela apresentação homogênea de informações quanto pelas técnicas de interação também homogêneas. A *Integração de Controle* "é responsável por prover serviços e funcionalidades básicas às ferramentas e ao ambiente" (FALBO, 1998, p.35), de forma a permiter o funcionamento organizado do mesmo. A *Integração de Dados* "estabelece as formas como as ferramentas realizarão o tratamento das informações", incluindo o gerenciamento dos serviços básicos de armazenamento e o gerenciamento das estruturas de informação.

No entanto, ainda segundo Falbo (1998), é necessário considerar também a natureza semântica das informações. Por exemplo, tarefas e domínios complexos possuem também um conjunto de informações complexas. Sistemas de informação que apóiam essas tarefas e domínios não possuem uma base de conhecimento capaz de gerar interpretações adequadas das informações presentes, assumindo que o usuário possui este conhecimento. Em ambientes de desenvolvimento de software, o conhecimento deve ser deslocado para o sistema, ou seja, é necessário *Integração de Conhecimento* (FALBO, 1998).

Segundo Falbo, Mian e Natali (2001), conhecimento e dados devem estar disponíveis e representados de forma única no ambiente de desenvolvimento de software, evitando redundâncias e inconsistências de informações entre ferramentas. Nesse contexto, as ontologias são especificamente importantes, pois são utilizadas na estruturação de bases de conhecimento e no compartilhamento de dados, assegurando a interoperabilidade entre ferramentas (OLIVEIRA, 1999).

Desta forma, as seções a seguir descrevem as duas ontologias elaboradas para possibilitar a integração de dados e, se possível, de conhecimento das ferramentas selecionadas no capítulo anterior. É importante ressaltar que essas ontologias não chegaram a ser codificadas em um ambiente que permitisse a verificacão da correcão dos axiomas definidos.

5.2 PROPÓSITO DAS ONTOLOGIAS

As ontologias de Gerência de Requisitos e de Gerência de Projeto têm como propósito definir um vocabulário comum, independente de ferramenta ou de domínio de aplicação específico, que identifique os elementos que precisam ser conhecidos na comunicação de dados entre as ferramentas que apóiam essas áreas do processo de desenvolvimento de software. Tal vocabulário também deve ser compartilhado pelos participantes do projeto de software que estão de alguma forma envolvidos com essas àreas de processo, pois ferramentas distintas podem conter funcionalidades semelhantes e é preciso compreender o fluxo de dados entre as ferramentas.

De forma mais detalhada, um conjunto de ferramentas apóia o processo de desenvolvimento de software. Por vezes, até mesmo duas ou mais ferramentas podem ser necessárias para apoiar uma única área de processo e, mesmo quando ferramentas apóiam áreas de processo diferentes, elas podem conter funcionalidades e/ou conceitos equivalentes. Por exemplo, duas ferramentas que apóiam, respectivamente, a Gerência de Requisitos e a Gerência de Projeto podem possibilitar o registro de tarefas a serem executadas. Na ferramenta de Gerência de Projeto, a funcionalidade pode aparecer como “Cadastrar Tarefas”,

enquanto que na ferramenta de Gerência de Requisitos, a funcionalidade pode ser “Registrar Pendências”. Ao registrar-se uma pendência na ferramenta de Gerência de Requisitos, deseja-se que este dado seja propagado e conste como uma tarefa na ferramenta de Gerência de Projeto. O propósito das ontologias aqui propostas é justamente definir um vocabulário comum para o qual os conceitos adotados nas diferentes ferramentas possam ser mapeados

5.3 CONSIDERAÇÕES SOBRE CAPTURA E FORMALIZAÇÃO

Considerando o propósito estabelecido para as ontologias, o ideal seria utilizar a linguagem OWL para representação das mesmas, uma vez que um arquivo XML (linguagem comumente utilizada para troca de dados) poderia ser diretamente gerado e utilizado para a integração das ferramentas. No entanto, foi utilizado o subconjunto da UML proposto por Mian (2003), que é apresentado no Anexo B, para representação do modelo conceitual das ontologias; linguagem natural para definição dos conceitos, relações e restrições; e lógica de primeira ordem para formalização da ontologia em termos de predicados e axiomas. Esta abordagem traz a vantagem de combinar elementos que facilitam o entendimento da ontologia pelos especialistas do domínio, os quais precisam validar os elementos conceituais capturados, com elementos que possibilitam a interpretação computacional, condição necessária à integração entre sistemas/ferramentas. Nesta dissertação, não há discussão extensiva das vantagens de se utilizar uma ou outra linguagem para a formalização de ontologias, porque ferramentas para edição de ontologias podem ser utilizadas para derivar automaticamente do modelo conceitual parte das restrições da ontologia, qualquer que seja a linguagem desejada para formalização (FALBO, 1998). Como trabalho futuro, será gerada a representação de ambas as ontologias em OWL.

5.4 ONTOLOGIA DE GERÊNCIA DE REQUISITOS

A ontologia de Gerência de Requisitos deve possibilitar que as seguintes questões de competência sejam respondidas:

- Pergunta 1: Quais são os critérios para a identificação de fornecedores de requisitos?

- Pergunta 2: Quais são os critérios para avaliação dos requisitos?

- Pergunta 3: Quais são os critérios para aceitação dos requisitos?

- Pergunta 4: Quais são os requisitos (técnicos e não técnicos, do cliente e também os estabelecidos pela organização desenvolvedora) associados ao projeto?

- Pergunta 5: Quais foram as fontes (principal e complementares) de um determinado requisito?

- Pergunta 6: Como um determinado requisito foi alterado ao longo do tempo?

- Pergunta 7: Qual a justificativa para uma determinada alteração de requisito?

- Pergunta 8: Quais requisitos que fazem parte de uma determinada versão da especificação de requisitos?

- Pergunta 9: Quem são os profissionais do cliente e da organização desenvolvedora comprometidos com um determinado requisito?

- Pergunta 10: Os requisitos atendem aos critérios estabelecidos para sua avaliação e aceitação?

- Pergunta 11: Quais problemas foram encontrados na avaliação dos requisitos?

- Pergunta 12: Quais artefatos do projeto foram avaliados de acordo com a consistência com os requisitos?

- Pergunta 13: Quais inconsistências foram encontradas em um artefato com relação aos requisitos?

- Pergunta 14: Quais são os níveis da matriz de rastreabilidade?

- Pergunta 15: Quais elementos de um mesmo nível da matriz de rastreabilidade possuem relações de dependência?

- Pergunta 16: Quais elementos de diferentes níveis da matriz de rastreabilidade possuem relações de dependência?

Para que as questões 4, 5 e 9 possam ser respondidas, é necessário definir 1) o escopo do projeto em termos de produtos a serem gerados e seus respectivos requisitos, além dos 2) profissionais envolvidos no projeto, tanto por parte da organização que solicitou o projeto, quanto por parte da organização empreendedora. A Figura 9 apresenta o modelo conceitual que trata desses aspectos. Os conceitos e relações são definidos nos Quadros 10 e 11, respectivamente. Axiomas epistemológicos, aqueles que descrevem as restrições impostas pela forma de estruturação dos conceitos, não serão representados, pois podem ser derivados automaticamente por uma ferramenta de edição de ontologias. Axiomas adicionais não se mostraram necessários neste contexto.

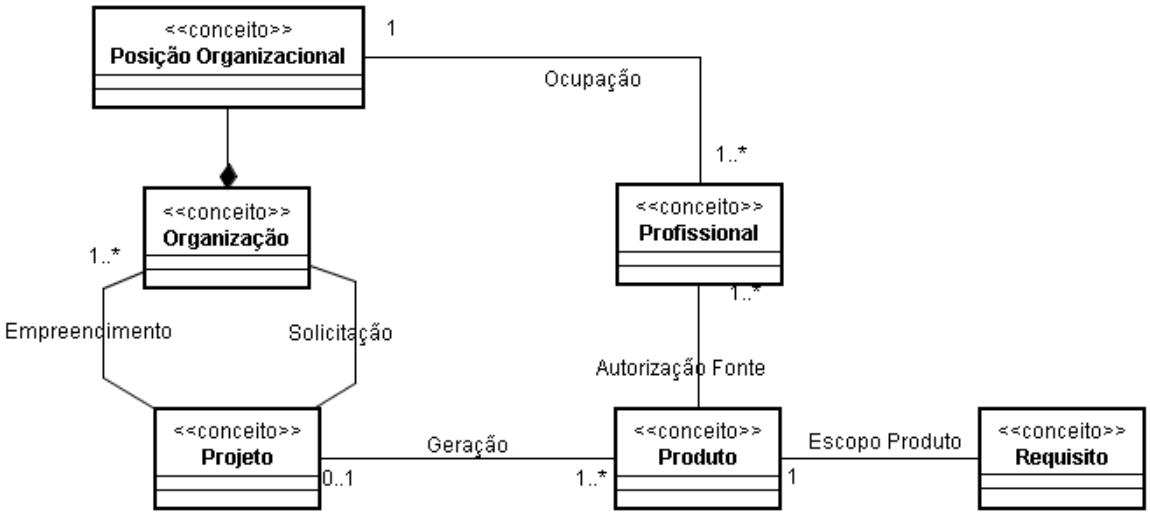


Figura 9 – Escopo do Projeto e Profissionais Envolvidos

CONCEITO	PREDICADO	DEFINIÇÃO
Organização	*Organizacao (org)*	Grupo de pessoas trabalhando em conjunto para o cumprimento de uma missão, onde a missão de uma organização define o seu propósito dentro do sistema econômico ou social (VILLELA, 2004).
Posição Organizacional	*PosOrganizacional (posorg)*	Especificação de um perfil profissional necessário para que a organização funcione de forma eficiente e econômica [baseado em (VILLELA, 2004).
Produto	*Produto(prod)*	Qualquer serviço ou saída tangível que é resultado de um processo, intencionado para ser entregue a um cliente ou usuário final (SEI, 2002).
Profissional	*Profissional (prof)*	Pessoa que atua na execução de uma ou mais atividades necessárias ao cumprimento da missão da organização (VILLELA, 2004).
Projeto	*Projeto (proj)*	Empreendimento de duração limitada, realizado para criar um produto único, cuja elaboração é progressiva (MPS.Br, 2007).
Requisito	*Requisito (req)*	Propriedade, restrição ou condição que determina como o produto será fornecido.

Quadro 10 – Escopo do Projeto e Profissionais Envolvidos: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Autorização Fonte	*AutorizaçãoFonte (prof, prod)*	O profissional **prof** está autorizado a fornecer requisitos para o produto **prod**
Composição Organização	*CompOrg (posorg, org)*	A posição organizacional **posorg** é uma das posições organizacionais que compõem a organização **org**
Escopo Produto	*Escopo (req, prod)*	O requisito **req** define o escopo do produto **prod**
Empreendimento	*Empreendimento (proj, orgsoft)*	O projeto **proj** é um empreendimento da organização **orgsoft**
Geração	*Geração (proj, prod)*	O projeto **proj** tem como objetivo gerar o produto **prod**
Ocupação	*Ocupação (prof, posorg)*	O profissional **prof** ocupa a posição organizacional **posorg**
Solicitação	*Solicitação (proj, orgcli)*	O projeto **proj** foi solicitado pela organização **orgcli**

Quadro 11 – Escopo do Projeto e Profissionais Envolvidos: Relações

As questões 14, 15 e 16 se referem à matriz de rastreabilidade, cujo modelo conceitual é apresentado na Figura 10. A matriz de rastreabilidade é formada por um conjunto de itens que compõem o produto (não necessariamente apenas requisitos), cada um associado a um determinado nível da matriz. As relações entre esses itens permitem a análise do impacto da alteração de um determinado requisito, como também de qualquer outro item da matriz. Os conceitos referentes à matriz de rastreabilidade, suas relações e os respectivos axiomas são

apresentados nos Quadros 12, 13 e 14. A partir deste ponto até o final do capítulo, os conceitos já previamente definidos serão apresentados com fundo cinza. Por exemplo, os conceitos Produto e Requisito, representados em cinza na Figura 10, foram mencionados pela primeira vez na Figura 9 e, conseqüentemente, já foram definidos anteriormente.

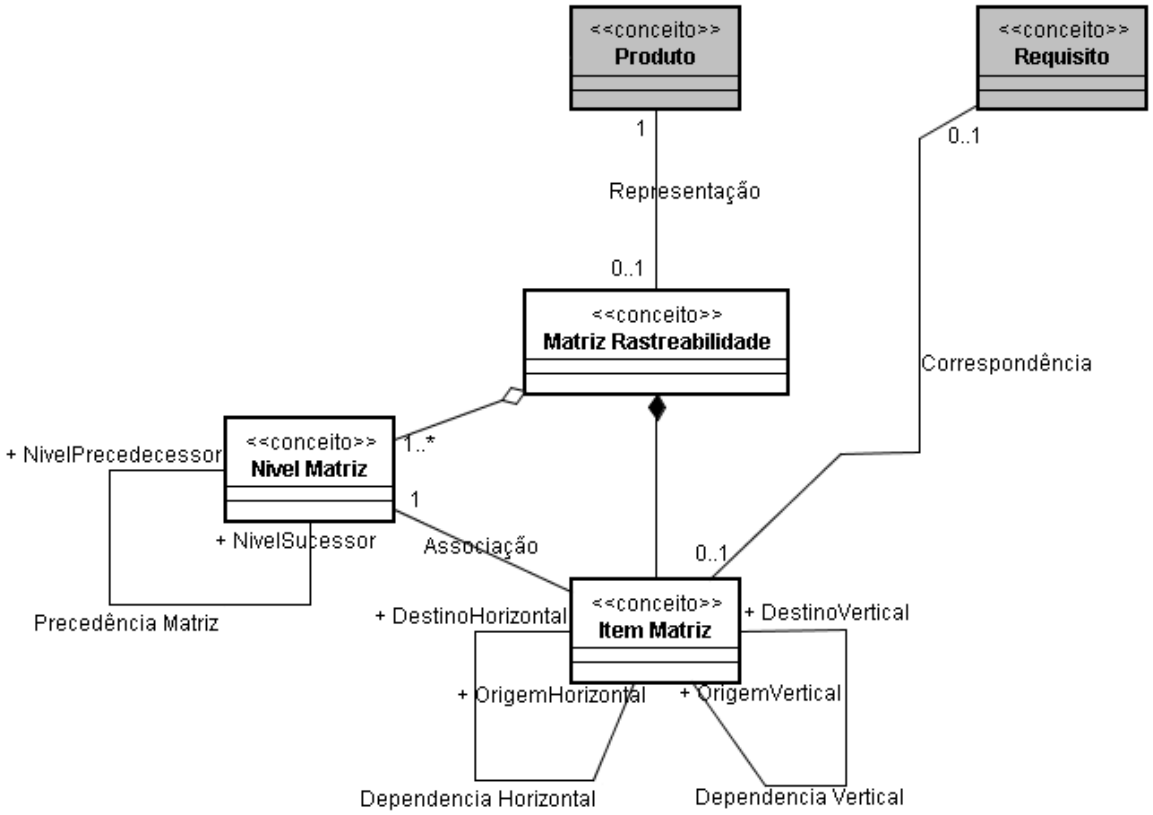


Figura 10 – Matriz de Rastreabilidade

RELAÇÃO	PREDICADO	DESCRIÇÃO
Item Matriz	*ItemMatriz (imat)*	Referência a um artefato ou parte de um artefato (por exemplo, requisito) que pode ser indivualmente identificada.
Matriz Rastreabilidade	*MatrizRastreabilidade (mr)*	Matriz que registra a relação entre artefatos ou partes de artefatos unicamente identificadas, podendo também registar a relação destes elementos com o ambiente (PRESSMAN, 2001).
Nível Matriz	*NivelMatriz (nmat)*	Nível de decomposição do produto, o qual pode representar desde diferentes níveis de requisitos a código fonte, passando por documentos intermediários, como, por exemplo, design e casos de teste (SEI, 2002).

Quadro 12 – Matriz de Rastreabilidade: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Associação	*Associação (imat, nmat)*	O item matriz **imat** está associado ao nível matriz **nmat**
Composição Conteúdo Matriz	*CompConteúdoMatriz (imat, mr)*	O item matriz **imat** é uma parte da matriz de rastreabilidade **mr**
Composição Estrutura Matriz	*CompEstruturaMatriz (nmat, mr)*	O nível matriz **nmat** é um dos níveis que definem a estrutura da matriz de rastreabilidade **mr**
Correspondência	*Correspondência (imat, req)*	O item matriz **imat** corresponde ao requisito **req**
Dependência Horizontal	*OrigemHorizontal (imat1, imat2)*	O item matriz **imat1** influenciou a definição do item matriz **imat2**, possuindo ambos o mesmo nível de abstração
	DestinoHorizontal (imat2, imat1)	O item matriz **imat2** depende do conteúdo do item matriz **imat1**, possuindo ambos o mesmo nível de abstração
Dependência Vertical	*OrigemVertical (imat1, imat2)*	O item matriz **imat1** foi utilizado como base para a definição do item matriz **imat2**, que é menos abstrato
	DestinoVertical(imat2, imat1)	O item matriz **imat2** foi definido com base no item matriz **imat1**, que é mais abstrato
Precedência Matriz	*NivelPredecessor (nmat1, nmat2)*	O nível matriz **nmat1** precede o nível matriz **nmat2**, ou seja, é mais abstrato
	NivelSucessor (nmat2, nmat1)	O nível matriz **nmat2** sucede o nível matriz **nmat1**, ou seja, é mais concreto
Representação	*Representação (prod, mr)*	O produto **prod** é representado pela matriz de rastreabilidade **mr**

Quadro 13 – Matriz de Rastreabilidade: Conceitos

NÚMERO	AXIOMA
A1	A relação de Precedência Matriz pode ser representada por dois predicados inversamente equivalentes
	$(\forall nmat_1,nmat_2)$ *NivelPredecessor*$(nmat_1,nmat_2) \leftrightarrow$ *NivelSucessor*$(nmat_2,nmat_1)$
A2	A relação de Precedência Matriz não é simétrica
	$(\forall nmat_1,nmat_2)$ *NivelPredecessor*$(nmat_1,nmat_2) \rightarrow \neg$ *NivelPredecessor*$(nmat_2,nmat_1)$
A3	A relação de Precedência de Matriz é transitiva
	$(\forall nmat_1,nmat_2,nmat_3)$ *NivelPredecessor*$(nmat_1,nmat_2) \wedge$ *NivelPredecessor*$(nmat_2,nmat_3) \rightarrow$ *NivelPredecessor*$(nmat_1,nmat_3)$
A4	A relação de Dependência Horizontal pode ser representada por dois predicados inversamente equivalentes
	$(\forall imat_1,imat_2)$ *OrigemHorizontal*$(imat_1,imat_2) \leftrightarrow$ *OrigemHorizontal*$(imat_2,imat_1)$
A5	A relação de Dependência Horizontal não é simétrica
	$(\forall imat_1,imat_2)$ *OrigemHorizontal*$(imat_1,imat_2) \rightarrow$ $\neg$ *OrigemHorizontal*$(imat_2,imat_1)$
A6	A relação de Dependência Horizontal é transitiva
	$(\forall imat_1, imat_2, imat_3)$ *OrigemHorizontal*$(imat_1, imatd_2) \wedge$ *OrigemHorizontal*$(imat_2, imat_3) \rightarrow$ *OrigemHorizontal*$(imat_1, imat_3)$
A7	A relação de Dependência Horizontal só pode ocorrer entre itens matriz associados ao mesmo nível matriz
	$(\forall imat_1,imat_2,nmat_1)$ *OrigemHorizontal*$(imat_1,imat_2) \wedge$ *Associação*$(imat_1,nmat_1) \rightarrow$ *Associação*$(imat_2,nmat_1)$

NÚMERO	AXIOMA
A8	A relação de Dependência Vertical pode ser representada por dois predicados inversamente equivalentes
	$(\forall imat_1, imat_2)\ OrigemVertical(imat_1, imat_2) \leftrightarrow OrigemVertical(imat_2, imat_1)$
A9	A relação de Dependência Vertical não é simétrica
	$(\forall imat_1, imat_2)\ OrigemVertical(imat_1, imat_2) \rightarrow \neg\ OrigemVertical(imat_2, imat_1)$
A10	A relação de Dependência Vertical é transitiva
	$(\forall imat_1, imat_2, imat_3)\ DestinoVertical(imat_1, imat_2) \wedge DestinoVertical(imat_2, imat_3) \rightarrow DestinoVertical(imat_1, imat_3)$
A11	A relação de Dependência Vertical só pode ocorrer entre itens matriz associados a diferentes níveis matriz
	$(\forall imat_1, imat_2, nmat_1)\ OrigemVertical(imat_1, imat_2) \wedge Associação\ (imat_1, nmat_1) \rightarrow \exists\ nmat2\ (NivelPredecessor(nmat_1, nmat_2) \wedge Associação(imat_2, nmat_2))$

Quadro 14 – Matriz de Rastreabilidade: Axiomas

A Figura 11 apresenta o modelo conceitual que complementa o apresentado na Figura 9, de forma a efetivamente responder as questões de competência 4 a 8. Os requisitos de um produto são, em geral, fornecidos pelos profissionais da organização solicitante e/ou empreendedora, mas também podem ser obtidos de algum artefato, como, por exemplo, documentação de um sistema legado. A Especificação de Requisitos reúne os requisitos em um único documento, os quais podem ser classificados em requisitos do cliente ou requisitos organizacionais e, de acordo com a especificidade técnica, em requisitos técnicos e não-técnicos. Os Requisitos técnicos, por sua vez, ainda podem ser classificados em funcionais e não funcionais. Os conceitos, relações e axiomas referentes à classificação e fonte dos requisitos, à substituição/ atualização de requisitos e da Especificação de Requisitos como um todo estão definidos nos Quadros 15, 16 e 17, respectivamente.

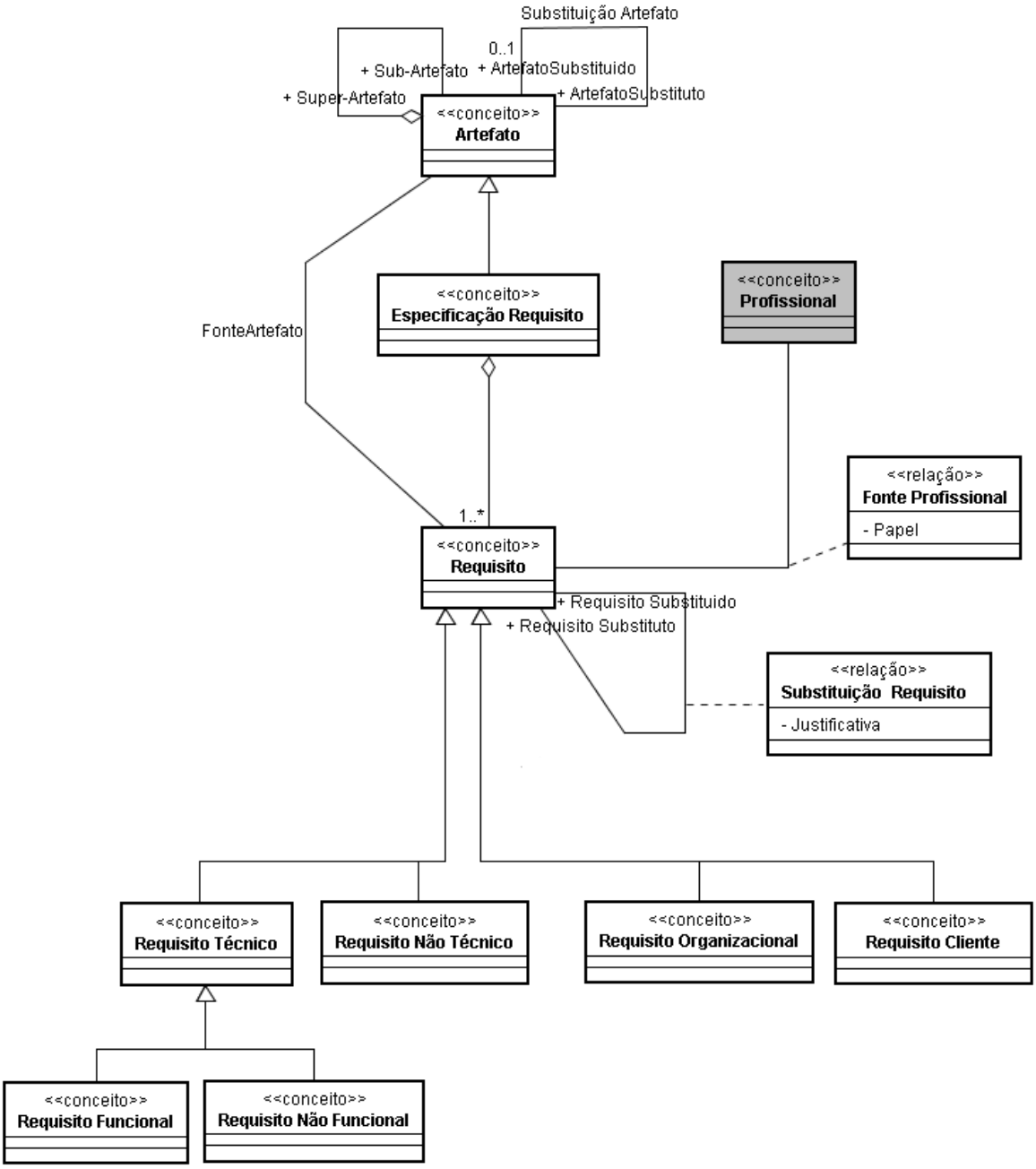


Figura 11 – Classificação, Fontes e Atualização de Requisitos

CONCEITO	PREDICADO	DESCRIÇÃO
Artefato	*Artefato (art)*	Um insumo para, ou um produto de, uma atividade, no sentido de ser um objeto de transformação da atividade (FALBO, 1998).
Especificação Requisito	*EspecificacaoRequisito (er)*	Artefato que reúne os requisitos de um produto, em formato eletrônico ou não, podendo ser sistematicamente revisado, avaliado e aprovado (SWEBOOK, 2004).
Requisito Cliente	*RequisitoCliente (reqcli)*	Requisito explicita ou implicitamente estabelecido para o produto pelo cliente. Requisitos implicitamente estabelecidos pelo cliente são propriedades que qualquer cliente razoável espera que o produto possua.
Requisito Funcional	*RequisitoFuncional (reqfun)*	Requisito técnico que estabelece funções que o produto deve ser capaz de executar, como o mesmo deve reagir a entradas especificas ou comportar-se em determinadas situações. Em alguns casos, pode também explicitamente declarar o que o produto não deve fazer (SOMMERVILLE, 2004).
Requisito Não Funcional	*RequisitoNFuncional (reqnfun)*	Requisito técnico que representa uma restrição sobre serviços ou funções oferecidos pelo sistema, ao invés de representar aspectos comportamentais. Entre outros exemplos, tem-se restrição de tempo e adequação a um padrão específico (SOMMERVILLE, 2004).
Requisito Não Técnico	*RequisitoNTecnico (reqntec)*	Requisito que representa uma determinação contratual, compromisso, condição ou termo que afeta como o produto será fornecido. Exemplos são a especificação dos artefatos a serem entregues e os critérios de saída nos marcos de entrega (SEI, 2002).
Requisito Organizacional	*RequisitoOrganizacional (reqorg)*	Requisito explicitamente estabelecido para o produto pela organização desenvolvedora.
Requisito Técnico	*RequisitoTecnico (reqtec)*	Requisito que representa uma propriedade que o produto a ser desenvolvido ou adaptado deve possuir para resolver um determinado problema (SWEBOOK, 2004).

Quadro 15 – Classificação, Fontes e Atualização de Requisitos: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Composição Artefato	*SuperArtefato (art2, art1)*	O artefato **art2** é um super-artefato do (ou é um todo, cuja uma das partes é o) artefato **art1**
	SubArtefato (art1, art2)	O artefato **art1** é um sub-artefato (ou é uma parte) do artefato **art2**
Composição Especificação	*CompEspecificação (req, er)*	O requisito **req** faz parte da especificação de requisitos **er**
Fonte Artefato	*FonteArtefato (art, req)*	O artefato **art** foi fonte do requisito **req**
Fonte Profissional	*FonteProfissional (prof, req, p)*	O profissional **prof** foi fonte do requisito **req**, desempenhando o papel **p**
Substituição Requisito	*Substituição de Requisitos (req1, req2, just)*	O requisito **req1** foi substituído pelo requisito **req2** de acordo com a justificativa **just**
	RequisitoSubstituto (req2, req1)	O requisito **req2** substituiu o requisito **req1**
	RequisitoSubstituído (req1, req2)	O requisito **req1** foi substituído pelo requisito **req2**
Substituição Artefato	*ArtefatoSubstituto (art2, art1)*	O artefato **art2** substituiu o artefato **art1**
	ArtefatoSubstituído (art1, art2)	O artefato **art1** foi substituído pelo **art2**

Quadro 16 – Classificação, Fontes e Atualização de Requisitos: Relações

NÚMERO	AXIOMA
A12	A relação de Substituição Artefato pode ser representada por dois predicados inversamente equivalentes
	$(\forall art_1,art_2)$ *ArtefatoSubstituido*$(art_1,art_2) \leftrightarrow$ *ArtefatoSubstituto*(art_2,art_1)
A13	A relação de Substituição Artefato não é simétrica
	$(\forall art_1,art_2)$ *ArtefatoSubstituido*$(art_1,art_2) \rightarrow \neg$ *ArtefatoSubstituido*(art_2,art_1)
A14	A relação de Substituição Artefato é transitiva
	$(\forall art_1,art_2,art_3)$ *ArtefatoSubstituido*$(art_1,art_2) \wedge$ *ArtefatoSubstituido*$(art_2,art_3) \rightarrow$ *ArtefatoSubstituido*(art_1,art_3)
A15	A relação de Composição Artefato pode ser representada por dois predicados inversamente equivalentes
	$(\forall art_1,art_2)$ *Superartefato*$(art_2,art_1) \leftrightarrow$ *Subartefato*(art_1,art_2)
A16	A relação de Composição Artefato não é simétrica
	$(\forall art_1,art_2,)$ *Subartefato* $(art_1,art_2) \rightarrow \neg$ *Subartefato*(art_2,art_1)
A17	A relação de Composição Artefato é transitiva
	$(\forall art_1,art_2,art_3)$ *Subartefato*$(art_1,art_2) \wedge$ *Subartefato*$(art_2,art_3) \rightarrow$ *Subartefato*(art_1,art_3)
A18	Macro-artefato é um artefato que não faz parte de nenhum outro artefato
	$(\forall art)$ *MacroArtefato(art)* $\leftrightarrow \neg(\exists art_1)$ *SuperArtefato(art1,art)*
A19	Artefato elementar é um artefato que não foi decomposto em outros artefatos
	$(\forall art)$ *ArtefatoElementar(art)* $\leftrightarrow \neg(\exists art_1)$ SubArtefato*(art1,art)*
A20	A relação de Substituição Requisito pode ser representada por dois predicados inversamente equivalentes
	$(\forall req_1,req_2)$ *RequisitoSubstituido*$(req_1,req_2) \leftrightarrow$ *RequisitoSubstituto*(req_2,req_1)
A21	A relação de Substituição Requisito não é simétrica
	$(\forall req_1,req_2)$ *RequisitoSubstituido*$(req_1,req_2) \rightarrow \neg$ *RequisitoSubstituido*(req_2,req_1)
A22	A relação de Substituição Requisito é transitiva
	$(\forall req_1,req_2,req_3)$ *RequisitoSubstituido*$(req_1,req_2) \wedge$ *RequisitoSubstituido*$(req_2,req_3) \rightarrow$ *RequisitoSubstituido*(req_1,req_3)
A23	Se um Profissional foi fonte de um Requisito, então está autorizado a ser fonte de requisito para o Produto, cujo escopo é definido pelo Requisito
	$(\forall prof,req,prod)$ *FonteProfissional(prof,req)* $\rightarrow$ *AutorizaçãoFonte(prof,prod)* $\wedge$ *Escopo(req,prod)*

Quadro 17 – Classificação, Fontes e Atualização de Requisitos: Axiomas

A Figura 12 apresenta os conceitos e relações necessários para que as questões de competência 2 e 3, além das questões de competência de 9 a 13 possam ser respondidas. A Especificação de Requisitos é um dos artefatos que definem um produto. Ao longo do projeto, esses artefatos são avaliados por profissionais de acordo com critérios pré-estabelecidos e as deficiências encontradas são registradas. Um critério de avaliação pode ser crítico para uma avaliação, o que significa que ele determina a aceitação ou não do artefato avaliado. Quando o artefato é aceito, os profissionais se comprometem com

conteúdo do mesmo. O Quadro 18, Quadro 19 e o Quadro 20 definem os conceitos, relações e axiomas referentes à Figura 12.

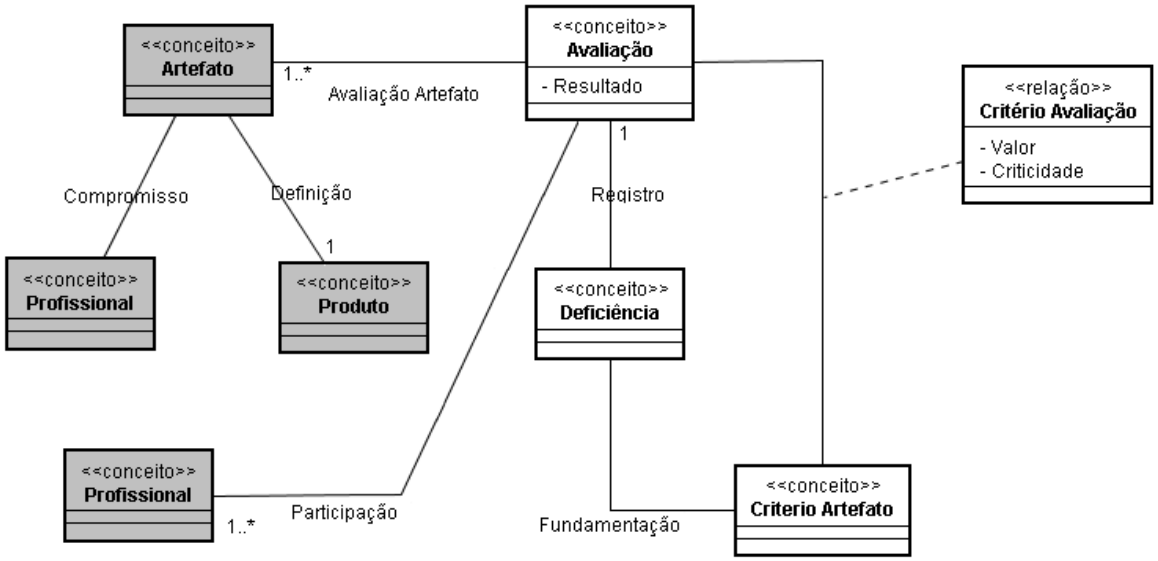


Figura 12 – Avaliação e Comprometimento com Artefatos

CONCEITO	PREDICADO	DESCRIÇÃO
Avaliação	*Avaliacao (ava)*	Avaliação de um ou mais artefatos por uma equipe de profissionais treinados, que podem fazer uso de um conjunto de critérios para determinar pontos fracos (deficiências) e pontos fortes no(s) artefato(s) avaliado(s).
Critério Artefato	*CritérioArtefato (criart)*	Critério a ser utilizado na avaliação de artefatos.
Deficiência	*Deficiência (def)*	Descrição de um problema encontrado durante uma avaliação, podendo estar relacionado ou não a critérios utilizados na avaliação.

Quadro 18 – Avaliação e Comprometimento com Artefatos: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Definição	*Definição (art, prd)*	O artefato **art** é um dos artefatos que define o produto **prd**
Avaliação Artefato	*AvaliaçãoArtefato (art, ava)*	O artefato **art** foi objeto da avaliação **ava**
Critério Avaliação	*CritérioAvaliação (ava, criart, val, cri)*	O critério artefato **criart** foi utilizado na avaliação **ava**, sendo **cri** a indicação de criticidade do critério para a aceitação do(s) artefato(s) envolvido(s) e **val** o valor atribuído ao(s) artefato(s) segundo o critério
Participação	*Participação(prof,ava)*	O profissional **prof** participou da avaliação **ava**.
Registro	*Registro (def, ava)*	A deficiência **def** foi registrada na avaliação **ava**
Fundamentação	*Fundamentação (criart, def)*	A deficiência **def** está fundamentada no critério artefato **criart**
Compromisso	*Compromisso (prof, art)*	O profissional **prof** está comprometido com o artefato **art**

Quadro 19 – Avaliação e Comprometimento com Artefatos: Relações

NÚMERO	AXIOMA
A24	Se uma Deficiência está fundamentada em um Critério Artefato, então existe uma Avaliação em que a Deficiência foi registrada, a qual tem como Critério Avaliação o Critério Artefato
	(∀ criart,def) Fundamentação(criart,def) → ∃ ava (Registro(def,ava) ∧ CritérioAvaliação(ava,criart,,*))*
A25	Se um Profissional está comprometido com um artefato, então o Profissional está alocado a uma Organização que é ou empreendedora ou solicitante do Projeto responsável por gerar o Produto definido pelo Artefato
	(∀ prof,art) Compromisso(prof, art) → ∃ org,proj,prod (Alocação(prof,org) ∧ (Empreendimento(proj,org) ∨ Solicitação(proj,org)) ∧ Geração(proj,prod) ∧ Definição(art,prod))

Quadro 20 – Avaliação e Comprometimento com Artefatos: Axiomas

A Figura 13 fornece os conceitos e relações relevantes tanto para responder a questão de competência 1, quanto para operacionalização dos criterios. Além dos critérios para a avaliação de artefatos, mencionados anteriormente, são necessários critérios para a seleção dos fornecedores de requisitos. Qualquer que seja o critério, existe uma escala que define os possíveis valores que uma entidade sendo considerada segundo o critério pode possuir. Tal escala pode ser numérica ou não numérica. As escalas não numéricas podem ser nominais ou ordinais, e são definidas por um conjunto de itens. Os Quadros 21, 22 e 23 fornecem os conceitos, relações e axiomas referentes à Figura 13.

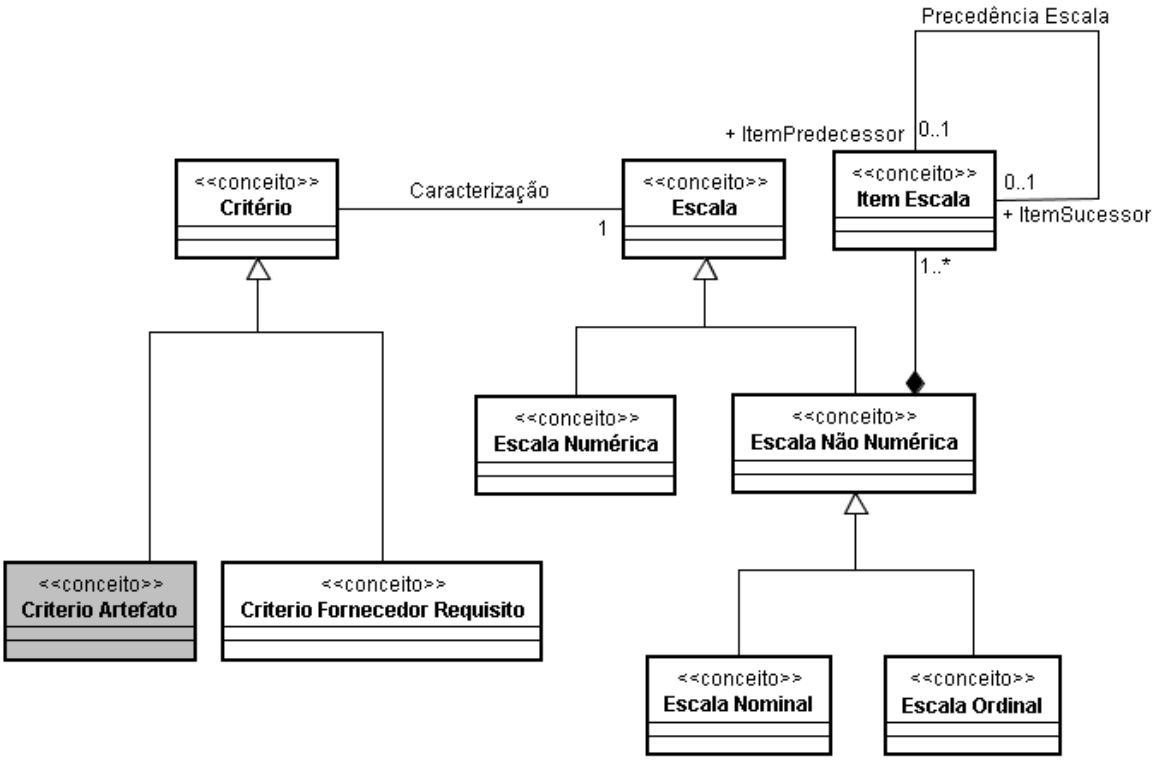


Figura 13 – Classificação de Critérios e Escalas

CONCEITO	PREDICADO	DEFINIÇÃO
Critério	Critério (cri)	Estabelecimento de uma propriedade a ser usada como fundamento ou base em uma verificação, validação, escolha ou decisão.
Critério Fornecedor Requisito	*CritérioFornecedorRequisito (crifornreq)*	Critério a ser utilizado na identificação de fornecedores de requisitos apropriados.
Escala	*Escala(esc)*	Definição de conjunto de valores que podem ser utilizados para representar o resultado da medição de um atributo ou propriedade.
Escala Numérica	*EscalaNumerica (escnum)*	Escala formada por um conjunto de valores numéricos.
Escala Não Numérica	*EscalaNaoNumerica (escnnum)*	Escala formada por um conjunto de valores não numéricos.
Escala Nominal	*EscalaNominal (escnom)*	Escala não numérica formada por um conjunto de categorias (ABNT, 2002a).
Escala Ordinal	*EscalaOrdinal (escord)*	Escala não numérica formada por um conjunto ordenado de valores (ABNT, 2002a).
Item Escala	*ItemEscala (itesc)*	Valor não numérico que possui um significado útil na representação do resultado da medição de um atributo ou propriedade.

Quadro 21 – Classificação de Critérios e Escalas: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Caracterização	*Caracterização (cri, esc)*	O critério **cri** é medido em termos da escala **esc**, que o caracteriza
Composição Escala	*CompEscala (itesc, escnnum)*	O item escala **itesc** faz parte da escala não numérica **escnnum**
Precedência Escala	*ItemPredecessor (itesc1, itesc2)*	O item escala **itesc1** precede o item escala **itesc2**
	ItemSucessor (itesc2, itesc1)	O item escala **itesc2** sucede o item escala **itesc1**

Quadro 22 – Classificação de Critérios e Escalas: Relações

NÚMERO	AXIOMA
A26	A relação de Precedência Escala pode ser representada por dois predicados inversamente equivalentes
	$(\forall itesc_1,itesc_2)\ ItemPredecessor(itesc_1,itesc_2) \leftrightarrow ItemSucessor(itesc_2,itesc_1)$
A27	A relação de Precedência Escala não é simétrica
	$(\forall itesc_1,itesc_2)\ ItemPredecessor(itesc_1,itesc_2) \rightarrow \neg\ ItemPredecessor(itesc_2,itesc_1)$
A28	A relação de Precedência Escala é transitiva
	$(\forall itesc_1,itesc_2,itesc_3)\ ItemPredecessor(itesc_1,itesc_2) \wedge ItemPredecessor(itesc_2,itesc_3) \rightarrow ItemPredecessor(itesc_1,itesc_3)$
A29	O valor atribuído a Valor em Critério Avaliação, quando o Critério Artefato é caracterizado por uma Escala Não Numérica, ou é nulo, ou é igual ao Item Escala que pertence à Escala Não Numérica que caracteriza o Critério Artefato
	$(\forall ava,criart,val,esc)\ CritérioAvaliação(ava,criart,val,*) \wedge Caracterização(criart,esc) \wedge EscalaNaoNumerica(esc) \rightarrow val = nulo \vee (\exists itesc)\ (CompEscala(itesc,esc) \wedge val = itesc)$
A30	Se existe uma relação de Precedência Escala entre dois Itens Escala, então ambos Itens Escala pertencem a mesma Escala Não Numérica.
	$(\forall itesc_1, itesc_2, escord)\ ItemPredecessor(itesc_1,itesc_2) \rightarrow (\exists escord)\ (CompEscala\ (itesc_1,escord) \wedge CompEscala\ (itesc_2,escord))$

Quadro 23 – Classificação de Critérios e Escalas: Axiomas

O Apêndice F apresenta o modelo conceitual completo da ontologia de Gerência de Requisitos.

5.5 ONTOLOGIA DE GERÊNCIA DE PROJETOS

Devido ao grande número de questões de competência identificadas, a ontologia de Gerência de Projetos foi dividida e é apresentada em seis partes: Escopo e Processo, Estimativas, Recursos Humanos, Informações, Monitoração e Riscos. Cada subseção a seguir contém as questões de competência, o modelo conceitual e as definições e restriçoes referentes a uma dessas cinco partes.

A) Escopo e Processo do Projeto

- Pergunta 1: Como os pacotes de trabalho que definem o escopo de um projeto estão organizados em termos de sub-pacotes e suas dependências?
- Pergunta 2: Como um determinado pacote de trabalho será obtido (desenvolvimento, aquisição ou reutilização)?
- Pergunta 3: Qual é o processo ou conjunto de atividades que implementam um pacote de trabalho?
- Pergunta 4: Como um processo (atividade) está organizado em termos de (sub-)atividades e suas dependências?
- Pergunta 5: Quais são as abordagens técnicas adotadas no projeto?
- Pergunta 6: Qual é o modelo de ciclo de vida que orienta a execução de um processo?
- Pergunta 7: Quais são os artefatos necessários para e gerados pela execução de uma atividade?

O escopo de um projeto é estabelecido em termos dos pacotes de trabalho previstos, definidos em uma Estrutura Analítica do Projeto (EAP), em Inglês *Working Breakdown Structure* (WBS), e as atividades necessárias para produzi-los podem ser organizadas em um processo. De uma forma geral, tanto o processo quanto atividades individuais podem estar baseados em uma abordagem técnica, como, por exemplo, desenvolvimento baseado em componentes. Como os conceitos de Processo e Atividade se relacionam da mesma forma com vários outros conceitos, eles podem ser considerados genericamente como uma Entidade Processual. A Figura 14 apresenta o modelo conceitual referente ao escopo do projeto e o processo utilizado para executá-lo.

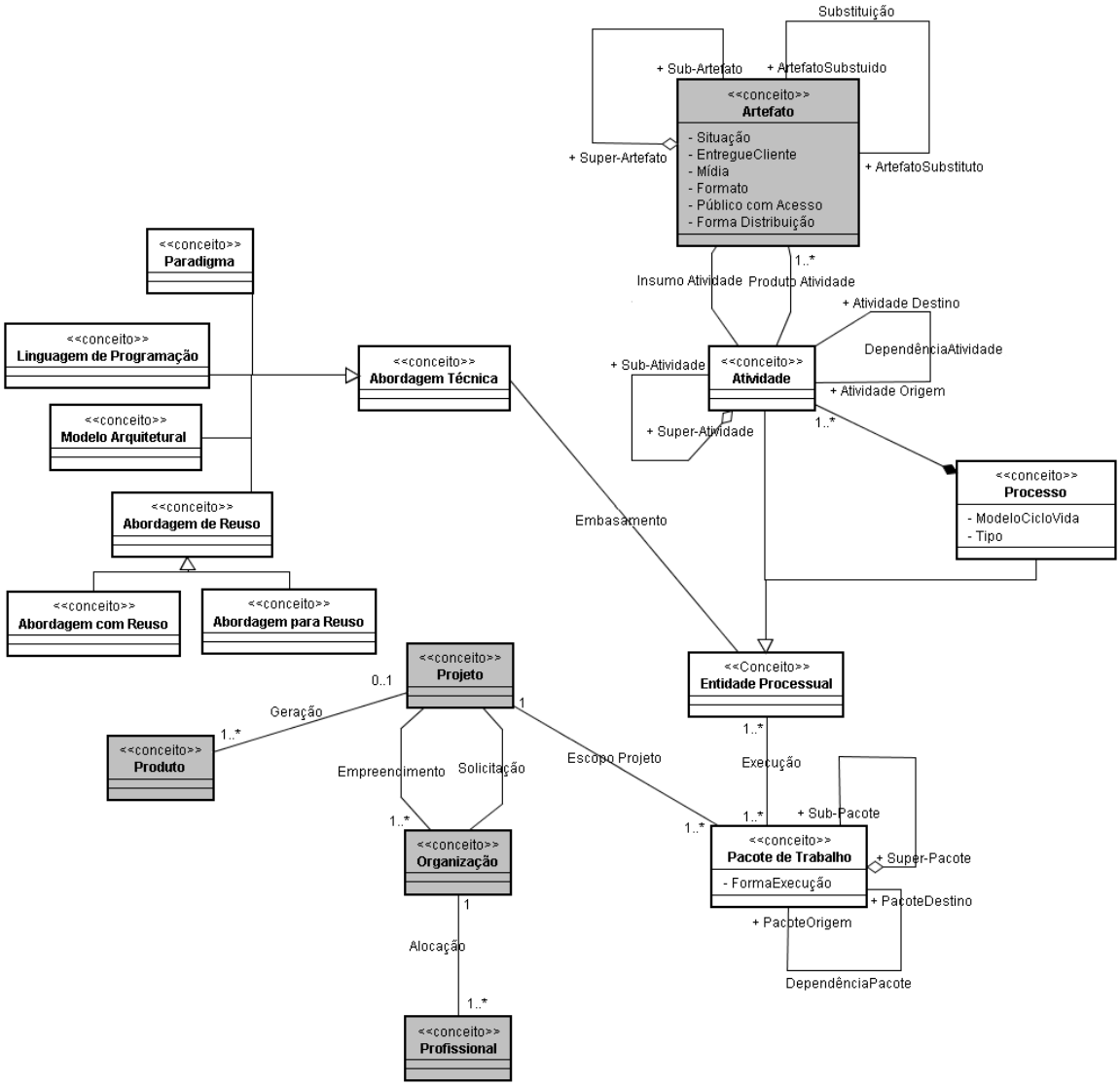


Figura 14 – Escopo e Processo do Projeto

Nos Quadros 24, 25 e 26 são apresentados os conceitos, relações e axiomas referentes ao modelo da Figura 14.

CONCEITO	PREDICADO	DEFINIÇÃO
Abordagem com Reuso	*AbordagemcomReuso (abordcreu)*	Abordagem de Reuso com o objetivo de reutilizar módulos disponíveis para o projeto.
Abordagem de Reuso	*AbordagemReuso (abordreu)*	Abordagem Técnica utilizada com a intenção de gerar artefatos reutilizáveis e, posteriormente, reutilizá-los.
Abordagem para Reuso	*AbordagemcomReuso (abordcreu)*	Abordagem de Reuso com o objetivo de gerar módulos para reutilização em outros projetos.
Abordagem Técnica	*AbordagemTécnica (abtec)*	Estratégia de alto nível a ser utilizada no desenvolvimento de produtos, o que inclui, por exemplo, decisões arquiteturais (SEI, 2002).
Atividade	*Atividade (at)*	Entidade processual que representa uma ação de transformação de artefatos de entrada (insumos) em artefatos de saída (produtos). Em função de sua natureza, as atividades podem ser classificadas em atividades de construção, atividades de gerência e atividades de avaliação da qualidade (FALBO, 1998).
Entidade Processual	*EntidadeProcessual (entproc)*	Entidade utilizada para a organização da execução do trabalho contratado em um projeto.
Linguagem de Programação	*LinguagemProgramação (lingproc)*	Abordagem Técnica que representa a linguagem artificial, composta por vocabulário e um conjunto de instruções, utilizada no desenvolvimento de um produto de software.
Modelo Arquitetural	*ModeloArquitetural (modarq)*	Abordagem Técnica que guia a estruturação dos componentes do software, incluindo os componentes propriamente ditos, as suas propriedades visíveis e o relacionamento entre eles. Exemplos de modelos arquiteturais são arquitetura cliente-servidor e de objetos distribuídos (J. FILHO, 2003).
Pacote de Trabalho	*PacoteTrabalho (pactrab)*	Entidade utilizada para a gerência da entrega do trabalho contratado em um projeto.
Paradigma	*Paradigma (parag)*	Abordagem Técnica contendo coleções de suposições, conceitos e práticas utilizados como guia para a execução de um processo ou atividade.
Processo	*Processo (proc)*	Entidade Processual composta por um conjunto estruturado de atividades, bem como toda a infra-estrutura envolvida na execução dessas atividades, o que inclui artefatos, procedimentos e recursos (FALBO, 1998).

Quadro 24 – Escopo e Processo do Projeto: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Composição Atividade	*SuperAtividade (at2, at1)*	A atividade **at2** é uma super-atividade da (ou é um todo, cuja uma das partes é a) atividade **at1**
	SubAtividade (at1, at2)	A atividade **at1** é uma sub-atividade (ou é uma uma parte) da atividade **at2**
Composição Pacote	*SuperPacote (pactrabt2, pactrab1)*	O pacote de trabalho **pactrab2** é um super-pacote do (ou é um todo, cuja uma das partes é o) pacote de trabalho **pactrab1**
	SubPacote (pactrab1, pactrab2)	O pacote de trabalho **pactrab1** é um sub-pacote (ou é uma parte) do pacote de trabalho **pactrab2**
Composição Processo	*ComposiçãoProcesso (at, proc)*	O processo **proc** é composto pela atividade **at**

RELAÇÃO	PREDICADO	DESCRIÇÃO
Dependência Atividade	*Pré-Atividade (at1, at2)*	A atividade **at1** antecede a atividade **at2**
	Pós-Atividade (at2, at1)	A atividade **at2** sucede a atividade **at1**
Dependência Pacote	*Pré-Pacote (pactrab1, pactrab2)*	O pacote de trabalho **pactrab1** antecede o pacote de trabalho **pactrab2**
	Pós-Pacote (pactrab2, pactrab1)	O pacote de trabalho **pactrab2** sucede o pacote de trabalho **pactrab1**
Embasamento	*Embasamento (entproc, abtec)*	A entidade processual **entproc** será executada com base na abordagem técnica **abtec**
Escopo Projeto	*EscopoProjeto (proj, pactrab)*	O projeto **proj** tem como tem como escopo o pacote de trabalho **pactrab**
Execução	*Execução (pactrab, entproc)*	O pacote de trabalho **pactrab** é executado através da entidade processual **entproc**
Insumo Atividade	*InsumoAtiv (art, at)*	O artefato **art** é insumo para a atividade **at**
Produto Atividade	*ProdutoAtiv (art, at)*	O artefato **art** é produto da atividade **at**

Quadro 25 – Escopo e Processo do Projeto: Relações

NÚMERO	AXIOMA
A31	A relação de Composição Pacote pode ser representada por dois predicados inversamente equivalentes
	$(\forall pactrab_1, pactrab_2)$ *SuperPacote*$(pactrab_2, pactrab_1) \leftrightarrow$ *SubPacote*$(pactrab_1, pactrab_2)$
A32	A relação de Composição Pacote não é simétrica
	$(\forall pactrab_1, pactrab_2)$ *SubPacote*$(pactrab_1, pactrab_2) \rightarrow \ulcorner$ *SubPacote*$(pactrab_2, pactrab_1)$
A33	A relação de Composição Pacote é transitiva
	$(\forall pactrab_1, pactrab_2, pactrab_3)$ *SubPacote*$(pactrab_1, pactrab_2) \wedge$ *SubPacote*$(pactrab_2, pactrab_3) \rightarrow$ *SubPacote*$(pactrab_1, pactrab_3)$
A34	Macro-pacote é um pacote de trabalho que não faz parte de nenhum outro pacote de trabalho
	$(\forall pactrab)$ *MacroPacote*$(pactrab) \leftrightarrow \neg(\exists pactrab_1)$ *SuperPacote*$(pactrab_1, pactrab)$
A35	Pacote elementar é um pacote de trabalho que não foi decomposto em outros pacotes de trabalho
	$(\forall pactrab)$ *PacoteElementar*$(pactrab) \leftrightarrow \neg(\exists pactrab_1)$ SubPacote$(pactrab_1, pactrab)$
A36	A relação de Dependência Pacote pode ser representada por dois predicados inversamente equivalentes
	$(\forall pactrab_1, pactrab_2)$ *PacoteOrigem*$(pactrab_1, pactrab_2) \leftrightarrow$ *Pacote Destino*$(pactrab_2, pactrab_1)$
A37	A relação de Dependência Pacote não é simétrica
	$(\forall pactrab_1, pactrab_2)$ *PacoteOrigem*$(pactrab_1, pactrab_2) \rightarrow \ulcorner$ *PacoteOrigem* $(pactrab_2, pactrab_1)$
A38	A relação de Dependência Pacote é transitiva
	$(\forall pactrab_1, pactrab_2, pactrab_3)$ *PacoteOrigem*$(pactrab_1, pactrab_2) \wedge$ *PacoteOrigem* $(pactrab_2, pactrab_3) \rightarrow$ *PacoteOrigem*$(pactrab_1, pactrab_3)$
A39	A relação de Composição Atividade pode ser representada por dois predicados inversamente equivalentes
	$(\forall at_1, at_2)$ *Pré-Atividade*$(at_1, at_2) \leftrightarrow$ *Pós-Atividade*(at_2, at_1)

NÚMERO	AXIOMA
A40	A relação de Composição Atividade não é simétrica
	$(\forall at_1,at_2)$ *Sub-Atividade*$(at_1,at_2) \rightarrow \ulcorner$ *Sub-Atividade*(at_2,at_1)
A41	A relação de Composição Atividade é transitiva
	$(\forall at_1,at_2,at_3)$ *Sub-Atividade*$(at_1,at_2) \wedge$ *Sub-Atividade*$(at_2,at_3) \rightarrow$ *Sub-Atividade*(at_1,at_3)
A42	Macro-atividade é uma atividade que não faz parte de nenhuma outra atividade
	$(\forall at)$ *MacroAtividade*$(at) \leftrightarrow \neg(\exists at_1)$ *SuperAtividade*(at_1,at)
A43	Atividade elementar é uma atividade que não foi decomposta em outras atividades
	$(\forall at)$ *AtividadeElementar*$(at) \leftrightarrow \neg(\exists at_1)$ SubAtividade(at_1,at)
A44	A relação de Dependência Atividade pode ser representada por dois predicados inversamente equivalentes
	$(\forall at_1,at_2)$ *Pré-Atividade*$(at_1,at_2) \leftrightarrow$ *Pós-Atividade*(at_2,at_1)
A45	A relação de Dependência Atividade não é simétrica
	$(\forall at_1,at_2)$ *Pré-Atividade*$(at_1,at_2) \rightarrow \ulcorner$ *Pré-Atividade*(at_2,at_1)
A46	A relação de Dependência Atividade é transitiva
	$(\forall at_1,at_2,at_3)$ *Pré-Atividade*$(at_1,at_2) \wedge$ *Pré-Atividade*$(at_2,at_3) \rightarrow$ *Pré-Atividade*(at_1,at_3)
A47	Se um artefato é insumo de uma atividade e contém sub-artefatos, então os sub-artefatos são insumos da mesma atividade
	$(\forall art, art_1, art_2,...,art_n,at,)$ *InsumoAtiv*$(art,at) \wedge$ *SubArtefato*$(art_1,art) \wedge ... \wedge$ *SubArtefato* $(art_n,art) \rightarrow$ *InsumoAtiv*$(art_1, at) \wedge ... \wedge$ *InsumoAtiv*(art_n,at)
A48	Se um artefato é produto de uma atividade e contém sub-artefatos, então os sub-artefatos são também produtos da mesma atividade
	$(\forall art, art_1, art_2,...,art_n,at,)$ *ProdutoAtiv*$(art,at) \wedge$ *SubArtefato*$(art_1,art) \wedge ... \wedge$ *SubArtefato* $(art_n,art) \rightarrow$ *ProdutoAtiv*$(art_1, at) \wedge ... \wedge$ *ProdutoAtiv*(art_n,at)

Quadro 26 – Escopo e Processo do Projeto: Axiomas

B) Estimativas do Projeto

- Pergunta 8: Quais atributos das atividades ou dos artefatos do projeto podem ser estimados?

- Pergunta 9: Quais são as estimativas feitas para os atributos das atividades ou dos artefatos do projeto?

- Pergunta 10: Quais são as estimativas (quantidade e percentual de utilização) de recursos necessários ao projeto?

- Pergunta 11: Quais são as estimativas de esforço do projeto?

- Pergunta 12: Quais são as estimativas de custo do projeto?

- Pergunta 13: Qual o cronograma estimado do projeto?

- Pergunta 14: Qual o orçamento estabelecido para o projeto?

- Pergunta 15: Qual o raciocínio que justifica o orçamento do projeto ou suas estimativas?

- Pergunta 16: Quais métodos e/ou técnicas foram utilizados na elaboração de uma estimativa ou de um conjunto de estimativas?

Os elementos mensuráveis de um projeto são os seus artefatos e as suas atividades, que podem ser estimados e mensurados em termos de tamanho e complexidade. Como base nas estimativas de tamanho e/ou de complexidade de artefatos ou atividades do projeto, todas as variáveis de projeto são estimadas: recurso, cronograma, orçamento, custo e esforço. Uma estimativa ou uma medição de uma variável de projeto pode estar associada a uma atividade específica ou ao projeto como um todo.

Independente de uma estimativa se referir a um elemento mensurável do projeto (artefato ou atividade) ou uma variável de projeto (recurso, cronograma, orçamento, custo e esforço) é importante registrar como ela foi obtida. Desta forma, deve ser registrado o procedimento utilizado, que pode ser um método, técnica ou diretriz.

Além disso, ao longo do projeto, valores reais referentes as estimativas dos elementos mensuráveis e das variáveis de projeto são coletados e, com base nesses valores, as estimativas ainda necessárias são atualizadas. A Figura 15 apresenta o modelo conceitual definido para responder as questões de competência referentes a estimativas. Os Quadros 27, 28 e 29 apresentam as definições dos respectivos conceitos, relações e axiomas.

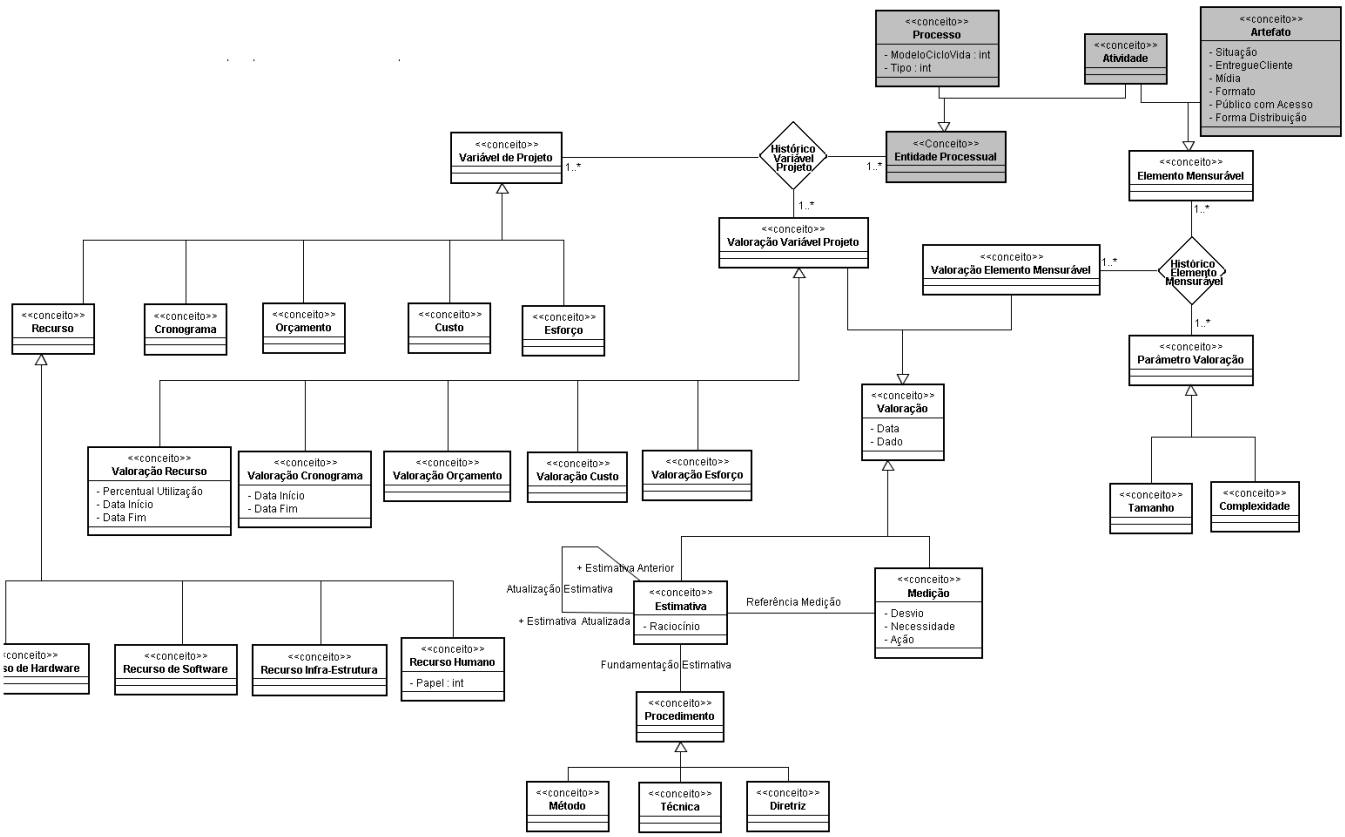


Figura 15 – Estimativas de Projeto

CONCEITO	PREDICADO	DEFINIÇÃO
Complexidade	*Complexidade (complex)*	Parâmetro Valoração que indica a dificuldade estimada/real de execução de uma determinada atividade ou de elaboração de um determinado artefato.
Cronograma	*Cronograma (cronog)*	Variável de Projeto referente às datas de início e fim de uma entidade processual.
Custo	*Custo (custo)*	Variável de Projeto referente ao valor monetário ou preço associado a uma entidade processual, considerando os recursos necessários para a sua execução (baseado em PMBOK, 2004).
Diretriz	*Diretriz (dirtz)*	Procedimento que visa estabelecer um padrão para a realização de atividades (FALBO, 1998).
Elemento Mensurável	*ElementoMensurável (emens)*	Elemento que faz parte do processo definido para o projeto e que pode ser medido quantitativamente.
Esforço	*Esforço (esf)*	Variável de Projeto referente a quantidade de unidades de mão-de-obra necessária para executar uma entidade processual (baseado em PMBOK, 2004).
Estimativa	*Estimativa (est)*	Valoração que representa uma avaliação antecipada da quantidade ou resultado provável.
Medição	*Medição (med)*	Valoração obtida através da ação de medir um parâmetro de valoração para um elemento mensurável ou medir uma variável de projeto para uma entidade processual.

CONCEITO	PREDICADO	DEFINIÇÃO
Método	*Método (met)*	Procedimento sistemático, que define passos e heurísticas para a execução de uma ou mais atividades (FALBO, 1998).
Orçamento	*Orçamento (orçam)*	Variável de Projeto referente ao valor monetário disponível para o projeto ou alocado a uma entidade processual
Parâmetro Valoração	*ParâmetroValoração (pval)*	Parâmetro utilizado na elaboração de uma estimativa ou na medição de um determinado elemento mensurável.
Procedimento	*Procedimento (proced)*	Conduta bem estabelecida e ordenada para a execução de uma atividade (FALBO, 1998).
Recurso	*Recurso (rec)*	Variável de Projeto referente a qualquer coisa que seja necessária para a realização de uma atividade, mas que não seja um insumo para a atividade, no sentido de não ser objeto de transformação por parte da atividade (FALBO, 1998).
Recurso Hardware	*Recurso Hardware (rechard)*	Recurso que especifica o hardware necessário para a execução de uma atividade. Por exemplo: computadores, multimídia (FALBO, 1998).
Recurso Software	*Recurso Software (recsoft)*	Recurso que especifica o software necessário para a execução de uma atividade, mas que não é incorporado ao produto da mesma (FALBO, 1998).
Recurso Humano	*Recurso Humano (rechum)*	Recurso que especifica o agente humano necessário para a execução de uma atividade. Por exemplo: engenheiro de software, programador, especialista de domínio (FALBO, 1998).
Recurso Infra Estrutura	*Recurso Infra Estrutura (recinfra)*	Recurso que especifica a instalação necessária para a execução de uma atividade. Por exemplo: acesso à internet, mesa para micro.
Tamanho	*Tamanho (tam)*	Parâmetro Valoração que indica o tamanho estimado/real de uma atividade ou artefato.
Técnica	*Técnica (tecn)*	Procedimento para a execução de uma atividade, que não a descreve em termos de passos (FALBO, 1998).
Valoração Cronograma	*ValoraçãoCronograma (valcronog)*	Valoração Variável Projeto que se aplica ao cronograma do projeto.
Valoração Custo	*ValoraçãoCusto (valcusto)*	Valoração Variável Projeto que se aplica ao custo do projeto.
Valoração Elemento Mensurável	*ValoraçãoElemMens (valelmens)*	Valoração de um elemento mensurável, o qual representa uma estimativa ou medição do mesmo com base no parâmetro de valoração utilizado.
Valoração Esforço	*ValoraçãoEsforço (valesf)*	Valoração Variável Projeto que se aplica ao esforço do projeto.
Valoração Orçamento	*ValoraçãoOrçamento (valorçam)*	Valoração Variável Projeto que se aplica ao orçamento do projeto.
Valoração Recurso	*ValoraçãoRecurso (valrec)*	Valoração Variável Projeto que se aplica a um recurso do projeto.
Valoração Variável Projeto	*ValoraçãoVariávelProj (valvarproj)*	Valoração de uma variável de projeto no contexto de uma determinada entidade processual, o qual representa uma estimativa ou medição da mesma.
Variável Projeto	*VariávelProjeto (varproj)*	Variável que caracteriza o projeto ao estabelecer uma restrição ou expectativa sobre o mesmo.

Quadro 27 – Estimativas do Projeto: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Atualização Estimativa	*EstimativaAnterior (estant, estatual)*	A estimativa **estant** foi atualizada através da estimativa **estatual**
	EstimativaAtualizada (estatual, estant)	A estimativa **estatual** é uma atualização da estimativa **estant**
Fundamentação Estimativa	*FundEstimativa (est, proced)*	A estimativa **est** foi realizada com base no procedimento **proced**
Histórico Elemento Mensurável	*HistóricoElemMens (emens, paramval, valemens)*	O parâmetro de valoração **paramval** do elemento mensurável **emens** obteve a Valoração Elemento Mensurável **valemens**
Histórico Variável Projeto	*HistóricoVariávelProj (varproj, entproc, valvarproj)*	A variável de projeto **varproj** para a entidade processual **entproc** obteve a Valoração Variável Projeto **valvarproj**
Referência Medição	*ReferênciaMedição (med, est)*	A medição **med** se refere à estimativa **est**

Quadro 28 – Estimativas do Projeto: Relações

NÚMERO	AXIOMA
A49	A relação de Atualização Estimativa pode ser representada por dois predicados inversamente equivalentes
	$(\forall$ *estant,estatual) EstimativaAnterior(estant,estatual)* $\leftrightarrow$ *EstimativaAtualizada (estatual,estant)*
A50	A relação de Atualização Estimativa não é simétrica
	$(\forall$ *estant,estatual) EstimativaAnterior(estant,estatual)* $\rightarrow$ $\neg$ *EstimativaAnterior(estatual,estant)*
A51	A relação de Atualização Estimativa é transitiva
	$(\forall est_1,est_2,est_3)$ $EstimativaAnterior(est_1,est_2) \wedge EstimativaAnterior(est_2,est_3) \rightarrow Estimativa\ Anterior(est_1,est_3)$
A52	Se uma medição se refere a uma estimativa, ambas são Valoração Variável Projeto e se relacionam com as mesmas instâncias de Variável Projeto e Entidade Processual, ou ambas são Valoração Elemento Mensurável e se relacionam com as mesmas instâncias de Elemento Mensurável e Parâmetro Valoração
	$(\forall$ *est,med) ReferênciaMedição(med,est)* $\rightarrow (\exists varproj_1,varproj_2,entproc_1,entproc_2)$ $(HistóricoVariávelProj(varproj_1,entproc_1,med) \wedge HistóricoVariávelProj(varproj_2,entproc_2,est) \wedge varproj_1 = varproj_2 \wedge entproc_1 = entproc_2) \vee (\exists emens_1, emens_2,paramval_1,paramval_2)$ $(HistóricoElemMens(emens_1,paramval_1,med) \wedge HistóricoElemMens(emens_2,paramval_2,est) \wedge emens_1 = emens_2 \wedge paramval_1 = paramval_2)$
A53	Variável de Projeto e Valoração Variável Projeto envolvidas em um mesmo Histórico Variável Projeto tem que ser compatíveis
	$(\forall entproc,varproj,valvarproj)$ $HistóricoVariávelProj(varproj,entproc,valvarproj) \rightarrow (Recurso(varproj) \wedge ValoraçãoRecurso(valvarproj)) \vee (Cronograma(varproj) \wedge ValoraçãoCronograma(valvarproj)) \vee (Orçamento(varproj) \wedge ValoraçãoOrçamento(valvarproj)) \vee (Custo(varproj) \wedge ValoraçãoCusto(valvarproj)) \vee (Esforço(varproj) \wedge ValoraçãoEsforço(valvarproj))$

Quadro 29 – Estimativas do Projeto: Axiomas

C) Recursos Humanos do Projeto

- Pergunta 17: Quais são o papel e as competências que definem um determinado recurso humano necessário à execução de um projeto?
- Pergunta 18: Quais são as competências de um profissional?
- Pergunta 19: Quais contratações e treinamentos foram solicitados no contexto de um projeto?
- Pergunta 20: Quais contratações e treinamentos foram solicitados de forma desvinculada de um projeto?
- Pergunta 21: Para quando contratações e treinamentos estão planejados?
- Pergunta 22: Quando contratações e treinamentos foram efetuados?
- Pergunta 23: Quais são os profissionais alvo de um determinado treinamento?
- Pergunta 24: Quais são as competências a serem aperfeiçoadas em um determinado treinamento?
- Pergunta 25: Quais perfis profissionais devem ser envolvidos em uma atividade do projeto, mesmo sem estarem alocados a ela, para que artefatos de saída mais adequados sejam gerados?
- Pergunta 26: Como os envolvimentos desejados podem ser especificados em termos de grau de interação, relevância e responsabilidade?
- Pergunta 27: Quais profissionais estão envolvidos em uma atividade do projeto?

O planejamento dos recursos humanos necessários a um projeto envolve mais do que estimar a quantidade de recursos e o seu período e percentual de utilização. É necessário estabelecer o conhecimento e as habilidades necessárias para a execução das atividades, de forma que profissionais possam ser selecionados para atuar como recursos humanos de um projeto conforme o conhecimento e as habilidades que possuem. Além disso, conhecimento e habilidades podem ser obtidos a partir de treinamento ou de contratação (CMMI, 2002), os quais podem ser simplesmente generalizados no conceito de capacitação. Um treinamento tem como objetivo que um grupo de profissionais desenvolva um conjunto de competências (conhecimento e/ou habilidades).

A demanda por treinamentos e contratações pode originar-se da necessidade de recursos humanos de um projeto específico ou de uma decisão no nível organizacional. Consequentemente, uma solicitação de capacitação pode estar vinculada a um projeto ou não. As solicitações desvinculadas são, em geral, preventivas.

Outro aspecto abordado nesta parte da ontologia de Gerência de Projeto é o planejamento e, conseqüentemente, registro do envolvimento de profissionais em atividades do projeto. O envolvimento de um profissional em uma atividade tem como objetivo apoiar os recursos humanos alocados à atividade, contribuindo para a adequação dos artefatos de saída. Os profissionais envolvidos não são diretamente responsáveis pela execução das atividades. Um exemplo ocorre quando o envolvimento de representantes da organização solicitante é planejado com o intuito desses fornecerem os requisitos do sistema a ser desenvolvido.

A Figura 16 apresenta o modelo conceitual referente à capacitação de recursos humanos e a Figura 17 é dedicada ao envolvimento de profissionais no processo definido para o projeto.

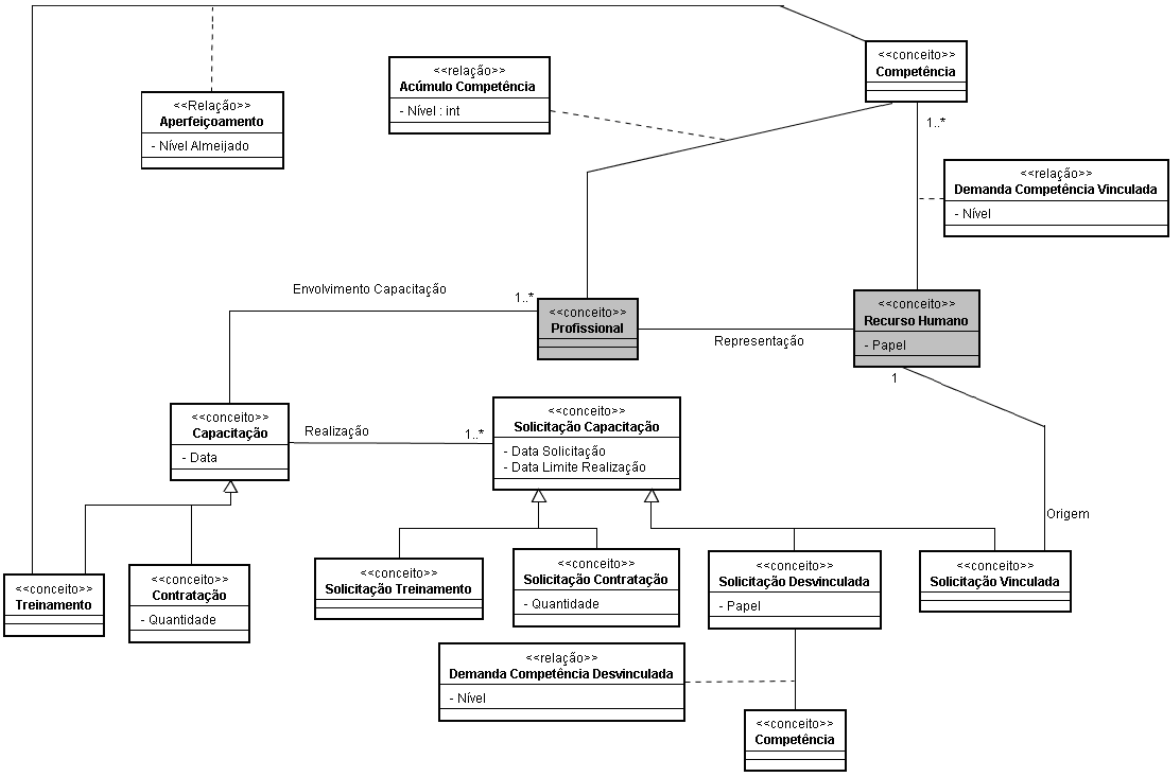


Figura 16 – Capacitação Profissional

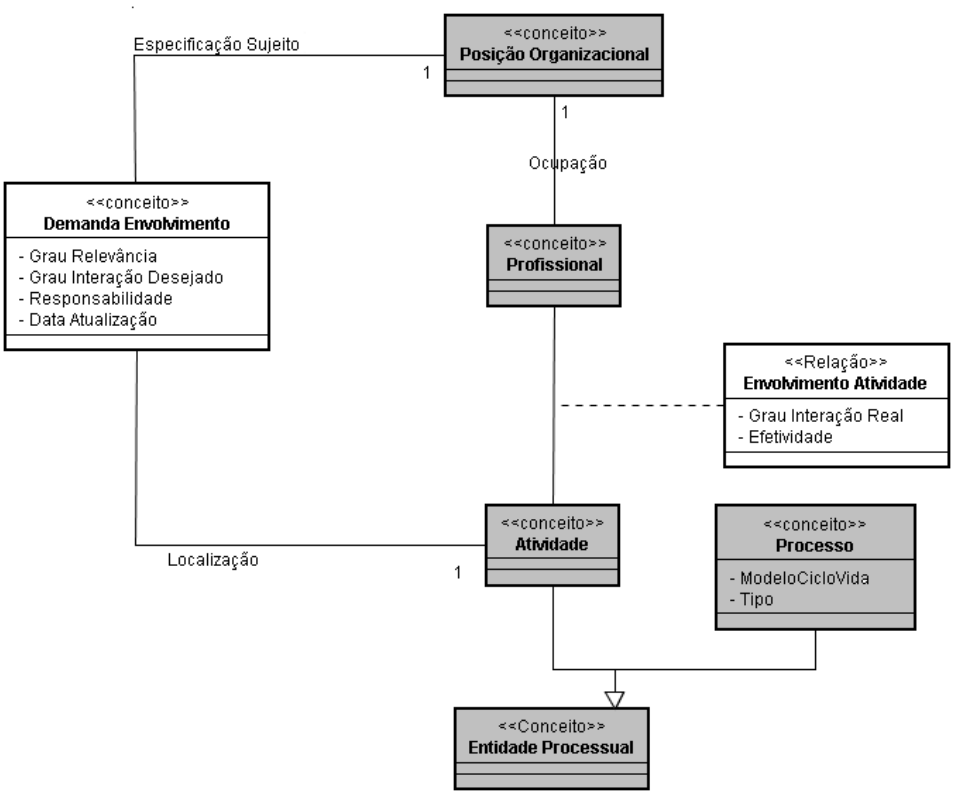


Figura 17 – Envolvimento Profissional

Os Quadros 30, 31 e 32 definem os conceitos, relações e axiomas referentes às Figuras 16 e 17.

CONCEITO	PREDICADO	DEFINIÇÃO
Capacitação	*Capacitação (capac)*	Atuação no quadro de profissionais da organização no sentido de adquirir uma competência específica e atender a demanda de recursos humanos de um projeto ou melhorar o desempenho da organização.
Competência	*Competência (compt)*	Conhecimento, habilidade ou experiência que torna o profissional capaz de executar atividades que envolvem algum grau de dificuldade (VILLELA, 2004)
Contratação	*Contratação (contrat)*	Capacitação que tem como objetivo disponibilizar novos profissionais para a organização.
Demanda Envolvimento	*DemandaEnvolv (demenvolv)*	Especificação do envolvimento necessário em uma determinada atividade de profissionais que ocupam uma determinada posição organizacional.
Solicitação Capacitação	*SolicCapacitação (solcap)*	Solicitação de capacitação da organização para atender a demanda de recursos humanos de um projeto ou melhorar o seu desempenho.

CONCEITO	PREDICADO	DEFINIÇÃO
Solicitação Contratação	*SolicContratação (solcontrat)*	Solicitação Capacitação referente a aquisição no mercado de trabalho de novos profissionais para a organização.
Solicitação Desvinculada	*SolicDesvinculada (soldesvinc)*	Solicitação Capacitação preventiva, preparatória e/ou no nível organizacional, não vinculada à demanda de recursos humanos de um projeto. Por exemplo, treinamento de um gerente de projeto na técnica de pontos por caso de uso.
Solicitação Treinamento	*SolicTreinamento (soltrein)*	Solicitação Capacitação referente ao desenvolvimento de novas competências em profissionais que já fazem parte da organização. Por exemplo: solicitação de treinamento na linguagem específica que será utilizada em um projeto ou é padrão na organização.
Solicitação Vinculada	*SolicVinculada (solvinc)*	Solicitação Capacitação vinculada à demanda de recursos humanos de um projeto.
Treinamento	*Treinamento (trein)*	Capacitação que tem como objetivo capacitar profissionais que já fazem parte da organização para o desempenho de um papel específico na organização ou em um projeto.

Quadro 30 – Recursos Humanos do Projeto: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Acúmulo Competência	*AcúmuloComp (prof, compt, niv)*	O profissional **prof** possui a competência **compt** no nível **niv**
Aperfeiçoamento	*Aperfeiçoamento (trein, compt, nivalm)*	O treinamento **trein** aperfeiçoa a competência **compt,** de forma a atingir o nível almejado **nivalm**
Demanda Competência Desvinculada	*DemandaCompDesvinc (soldesvinc, compt, niv)*	A solicitação desvinculada **soldesvinc** demanda a competência **compt** no nível **niv**
Demanda Competência Vinculada	*DemandaCompVinc (rechum, compt, niv)*	O recurso humano **rechum** precisa ter a competência **compt** no nível **niv**
Envolvimento Capacitação	*EnvolvimentoCapac (prof, capac)*	O profissional **prof** está envolvido na capacitação **capac**, com sujeito da contratação ou do treinamento
Especificação Sujeito	*EspecificacaoSujeito (demenv, posorg, graurel, grauint, resp)*	A demanda envolvimento **demenv** demanda o envolvimento de profissionais que ocupam a posição organizacional **posorg**, especificando grau de relevância **graurel**, grau de interação **grauint** e responsabilidade **resp**
Localização	*Localização (demenv, at)*	A demanda envolvimento **demenv** refere-se à atividade **at**
Ocupação	*Ocupação (prof, posorg)*	O profissional **prof** ocupa a posição organizacional **posorg**
Origem	*Origem (solvinc, rechum)*	A solicitação vinculada **solvinc** tem origem na especificação de recurso humano **rechum**
Realização	*Realização (solcapac, capac)*	A solicitação capacitação **solcapac** foi atendida através da capacitação **capac**
Representação	*Representação (prof, rechum)*	O profissional **prof** atua como recurso humano **rechum**

Quadro 31 – Recursos Humanos do Projeto: Relações

NÚMERO	AXIOMA
A54	Se um profissional está envolvido em um treinamento, então o treinamento adiciona uma competência ao profissional ou desenvolve (aumenta o nível de) uma competência já existente
	($\forall$ prof,capac,compt,nivalm) EnvolvimentoCapac (prof,capac) $\wedge$ Aperfeiçoamento (capac,compt, nivalm) $\rightarrow$ AcúmuloComp (prof,compt,nivalm)

Quadro 32 – Recursos Humanos do Projeto: Axiomas

D) Informações do Projeto

- Pergunta 28: Quais são a mídia, o formato, o público-alvo e a forma de distribuição de cada artefato do projeto?

- Pergunta 29: Qual roteiro foi utilizado para a elaboração de um determinado artefato do projeto?

- Pergunta 30: Quais diferentes visões de um determinado artefato do projeto são possíveis de acordo com o público alvo?

- Pergunta 31: Em que situação encontra-se um determinado artefato do projeto?

- Pergunta 32: Quais são os comprometimentos a serem obtidos para um determinado artefato?

- Pergunta 33: Quem está comprometido com o conteúdo de um artefato?

- Pergunta 34: Um artefato precisa ser novamente comprometido?

O planejamento das informações do projeto inclui a especificação de como os artefatos de entrada serão fornecidos, como os artefatos de saída serão produzidos e distribuídos para o público-alvo, como também se é necessário que os profissionais das organizações envolvidas se compremetam com o conteúdo dos artefatos. A Figura 18 apresenta o modelo conceitual referente às informações de projeto.

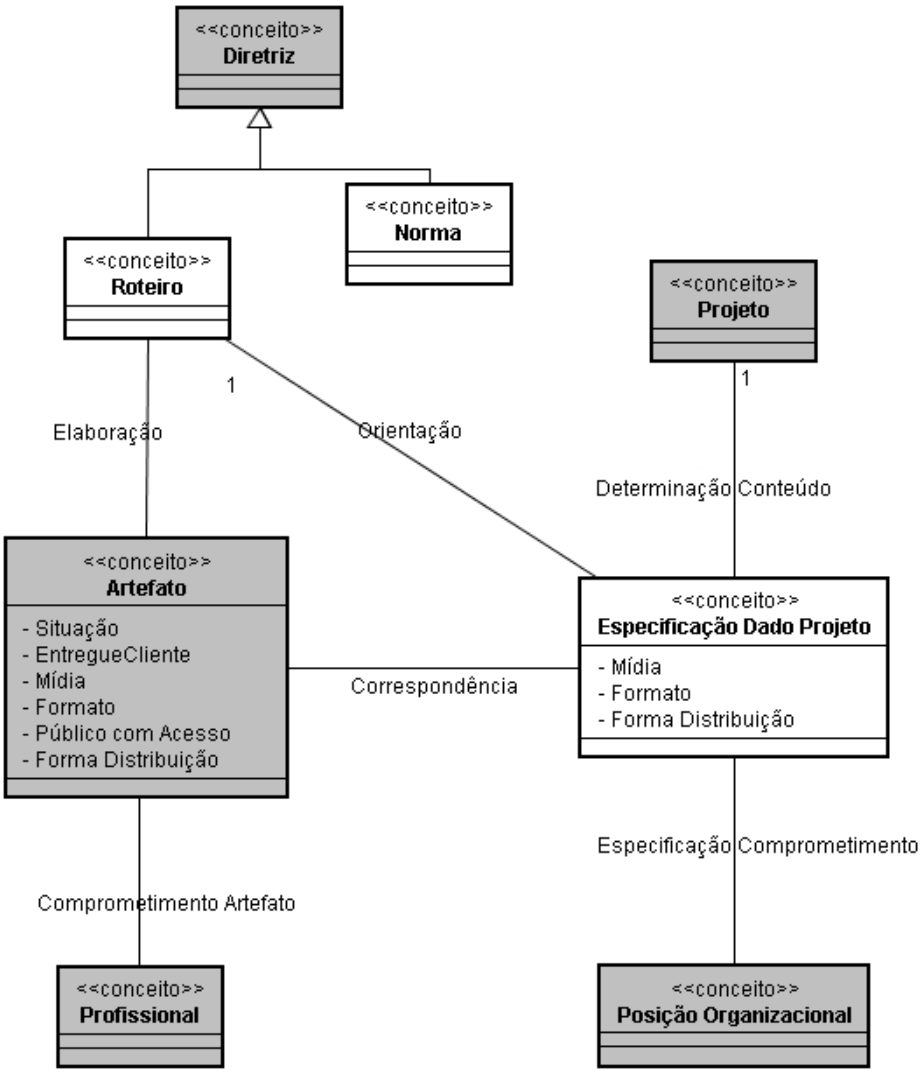


Figura 18 – Informações do Projeto

Os Quadros 33, 34 e 35 definem os conceitos, relações e axiomas referentes à Figura 18.

CONCEITO	PREDICADO	DEFINIÇÃO
Especificação Dado Projeto	*EspecDadoProjeto (dadoproj)*	Especificação de como um artefato deve ser tratado no contexto de um projeto específico, o que inclui elaboração, armazenamento, comprometimentos e distribuição.
Norma	*Norma (norma)*	Diretriz que visa estabelecer padrões para execução de atividades que não sejam de elaboração de documentos. Por exemplo: normas de programação (FALBO, 1998)
Roteiro	*Roteiro(rote)*	Diretriz para a elaboração de documentos. Por exemplo: roteiro de Plano de Projeto (FALBO, 1998)

Quadro 33 – Informações do Projeto: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Comprometimento Artefato	*ComprometimentoArtefato (prof,art)*	O profissional **prof** está comprometido com o artefato **art**
Correspondência	*Correspondência (art, dadoproj)*	O artefato **art** corresponde à especificação dado projeto ***dadoproj***
Determinação Conteúdo	*DeterminacaoConteúdo (dadoproj, proj, mid, form, pubalv)*	O projeto **proj** tem o conteúdo determinado pela especificação dado projeto **dadoproj**, a qual informa mídia **mid**, formato **form** e público alvo **pubalv** dos artefatos a serem elaborados segunfo a especificação
Especificação Comprometimento	*EspecComprometimento (posorg, dadoproj)*	A especificação dado projeto **dadoproj** determina que profissionais que ocupam a posição organizacional **posorg** se comprometam com os artefatos gerados segundo a especificação
Elaboração	*Elaboração (art, rot)*	O artefato **art** foi elaborado com base no roteiro **rot**
Orientação	*Orientação (rot, dadoproj)*	A especificação dado projeto **dadoproj** determina que os artefatos correspondentes à especificação sejam elaborados segundo o roteiro **rot**

Quadro 34 – Informações do Projeto: Relações

NÚMERO	AXIOMA
A55	Se um artefato corresponde a uma Especificação Dado Projeto de um projeto, então este artefato é insumo ou produto de uma atividade definida para o projeto
	(∀ art,dadoproj) Correspondencia (art,dadoproj) → (∃ proj) Conteúdo(dadoproj,proj) ∧ (∃ entproc,pactrab) (EscopoProjeto(proj,pactrab) ∧ Execução(pactrab,entproc) ∧ (InsumoAtiv(art,entproc) ∨ ProdutoAtiv(art,entproc) ∨ (∃ ativ) (ComposicaoProcesso (ativ,entproc) ∧ (InsumoAtiv(art,ativ) ∨ ProdutoAtiv(art,ativ)))))

Quadro 35 – Informações do Projeto: Axiomas

E) Riscos Associados ao Projeto

- Pergunta 35: Quais são os riscos do projeto?

- Pergunta 36: A que categoria de risco pertence um risco?

- Pergunta 37: Qual é a probabilidade e o impacto de um risco?

- Pergunta 38: Qual prioridade foi atribuída a um risco?

- Pergunta 39: Quais são os riscos que serão gerenciados no escopo do projeto?

- Pergunta 40: Qual é a situação de um risco?

Quando um projeto é estimado, ou até mesmo na fase de elaboração do escopo, os potenciais problemas associados ao projeto são analisados, os quais são chamados de riscos. Segundo Pressman (2001), “identificação de riscos é

uma tentativa sistemática de especificar problemas para o plano de projeto (estimativas, cronograma, recursos, etc..)". Esses riscos são categorizados com o objetivo de serem agrupados em possíveis causas. Além disso, possuem algumas características associadas, as quais são: probabilidade de ocorrência do risco no contexto do projeto, impacto da ocorrência do risco no projeto e prioridade do risco na análise dos riscos (PRESSMAN, 2001). O impacto da ocorrência do risco no projeto pode ser tanto positivo quanto negativo, e, no caso de ser negativo, é necessário definir um plano de mitigação e/ou contigência. Quanto à priorização, riscos com maior probabilidade e maior impacto têm maior prioridade na análise de riscos. Além disso, riscos com baixa probabilidade de ocorrência e baixo impacto geralmente não são gerenciados ao longo do projeto. A Figura 19 apresenta o modelo conceitual referente aos riscos do projeto, no qual a situação de um risco também é representada, para indicar se o risco já ocorreu ou se deixou de ser um risco porque a fase em que poderia ocorrer já foi concluída, entre outras situações.

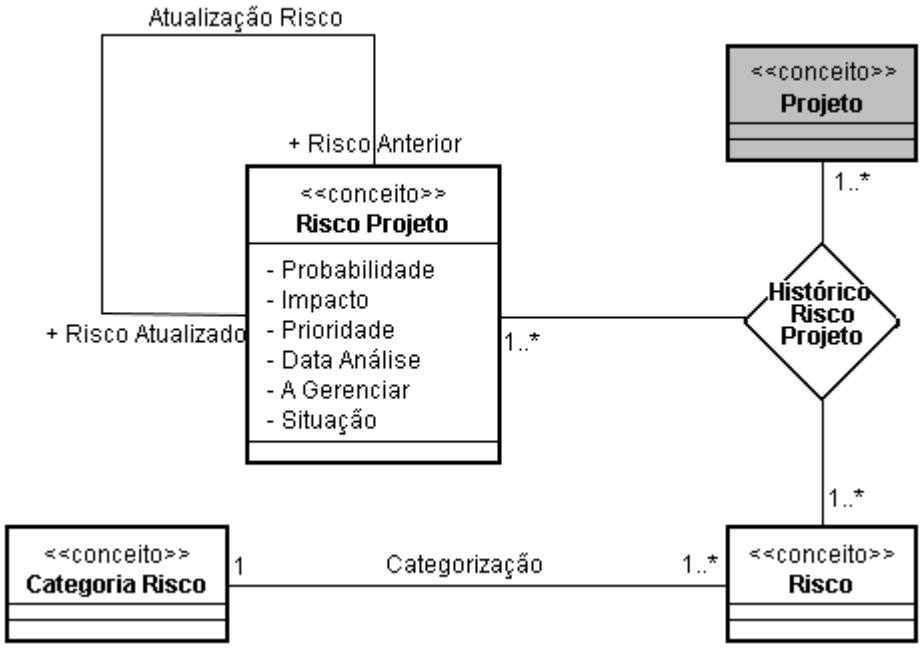


Figura 19 – Riscos Associados ao Projeto

Nos Quadros 36, 37 e 38, os conceitos, relações e axiomas referentes à Figura 19 são definidos.

CONCEITO	PREDICADO	DEFINIÇÃO
Risco	*Risco (risc)*	Evento que pode ocorrer ou não e, caso ocorra, pode ter um impacto negativo no projeto (PRESSMAN, 2001).
Risco Projeto	*RiscoProjeto (riscproj)*	Risco analisado no contexto de um projeto.
Categoria Risco	*CategoriaRisco (catrisc)*	Grupo de possíveis causas de riscos que facilita a identificação e organização dos mesmos. Um dos objetivos da categorização dos riscos é ajudar na consolidação de atividades de mitigação de riscos (SEI, 2002).

Quadro 36 – Riscos do Projeto: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Atualização Risco	*RiscoAnterior ($riscproj_1$, $riscproj_2$)*	A análise contida no risco do projeto **riscproj1** foi atualizada pela analise contida no risco do projeto **riscproj2**
	RiscoAtualizado ($riscproj_2$, $riscproj_1$)	A análise contida no risco do projeto **riscproj2** é uma atualização da análise contida no risco do projeto **riscproj1**
Categorização	*Categorização (risc, catrisc)*	O risco **risc** pertence à categoria de risco **catrisc**
Histórico Risco Projeto	*HistRiscoProjeto (risc, proj, riscproj)*	O risco **risc** é relevante no contexto do projeto **proj**, tendo sido analisado conforme descrito em **riscproj**

Quadro 37 – Riscos do Projeto: Relações

NÚMERO	AXIOMA
A56	A relação de Atualização Risco pode ser representada por dois predicados inversamente equivalentes
	$(\forall riscproj_1, riscproj_2)$ *RiscoAnterior*$(riscproj_1, riscproj_2) \leftrightarrow$ *RiscoAtualizado* $(riscproj_2, riscproj_1)$
A57	A relação Atualização Risco não é simétrica
	$(\forall riscproj_1, riscproj_2)$ *RiscoAnterior*$(riscproj_1, riscproj_2) \rightarrow$ $\neg$ *RiscoAnterior*$(riscproj_2, riscproj_1)$
A58	A relação de Atualização Risco é transitiva
	$(\forall riscproj_1, riscproj_2, riscproj_3)$ *RiscoAnterior*$(riscproj_1, riscproj_2) \wedge$ *RiscoAnterior*$(riscproj_2, riscproj_3) \rightarrow$ *RiscoAnterior*$(riscproj_1, riscproj_3)$
A59	Se um risco do projeto foi atualizado, então ambas as instâncias se referem ao mesmo projeto e mesmo risco
	$(\forall riscproj_1, riscproj_2)$ *RiscoAnterior* $(riscproj_1, riscproj_2) \rightarrow (\exists risc_1, risc_2, proj_1, proj_2)$ $(HistRiscoProjeto(risc_1, proj_1, riscproj_1) \wedge HistRiscoProjeto(risc_2, proj_2, riscproj_2) \wedge risc_1 = risc_2, \wedge proj_1 = proj_2)$

Quadro 38 – Riscos do Projeto: Axiomas

F) Monitoração do Projeto

- Pergunta 41: Qual são os dados reais referente às estimativas, especificações de dados de projeto, contratações e treinamentos planejados, envolvimentos de

profissionais em atividades, compromissos com artefatos e riscos identificados para o projeto?

- Pergunta 42: Qual é o desvio existente entre um dado real e um planejamento ou uma estimativa?

- Pergunta 43: Quais ações corretivas são necessárias no projeto em função de um desvio significativo entre o dado real e um planejamento ou estimativa?

- Pergunta 44: Quando ocorreram eventos de monitoração e controle no projeto?

- Pergunta 45: Quais profissionais participaram dos eventos de monitoração e controle do projeto?

- Pergunta 46: Quais outras ações foram determinadas nos eventos de monitoração e controle do projeto?

Após realizar as estimativas, é necessário acompanhá-las, comparando-as com os valores reais obtidos até o momento. Um exemplo é verificar se o custo atual do projeto está dentro da expectativa ou limitação imposta inicialmente. Para isto, são efetuadas monitorações em marcos de projeto e, eventualmente, em pontos de controle. Critérios de monitoração auxiliam na identificação de desvios significativos das estimativas e planos, os quais precisam ser tratados através de ações corretivas. Essas ações são agrupadas em um plano de ação e podem requerer, por exemplo, re-planejamento, estabelecimento de novos acordos ou inclusão de atividades de mitigação adicionais no plano de projeto (CMMI, 2002).

A Figura 20 apresenta o modelo conceitual relativo à monitoração de projeto e os Quadros 39, 40 e 41 definem os respectivos conceitos, relações e axiomas.

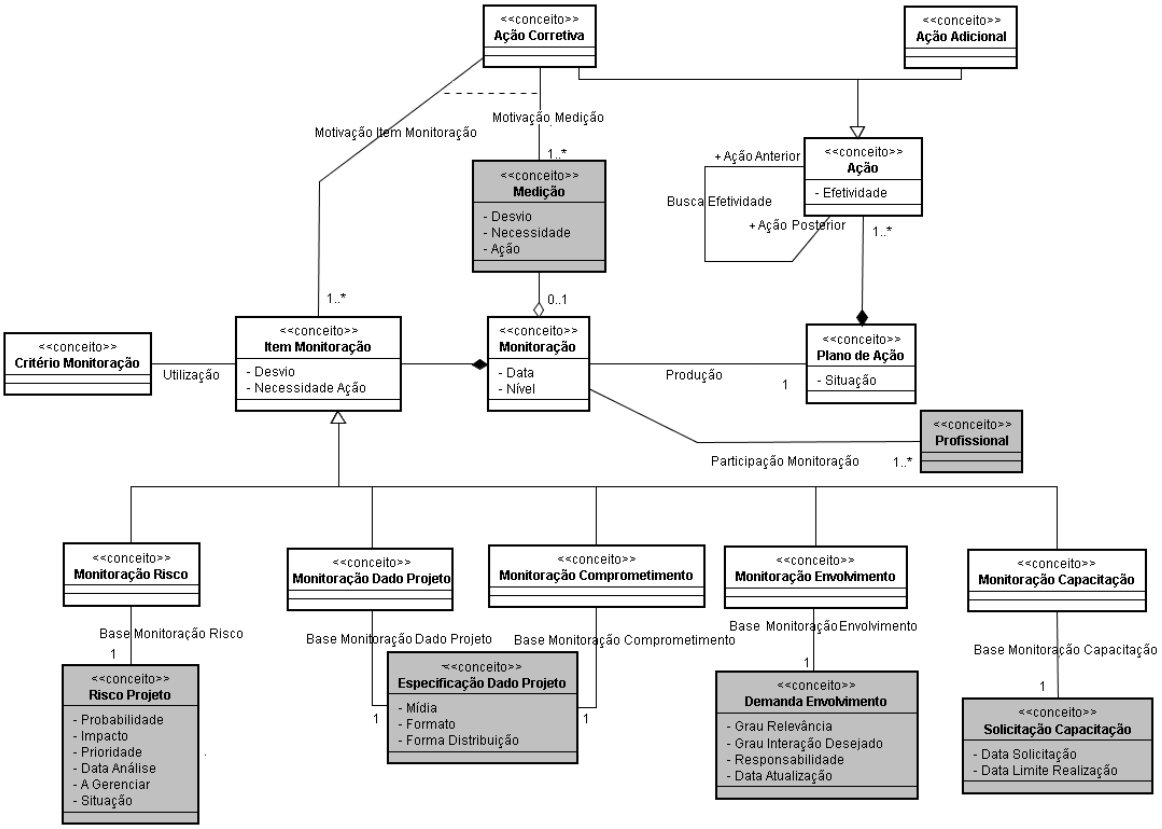


Figura 20 – Monitoração do Projeto

CONCEITO	PREDICADO	DEFINIÇÃO
Ação	*Ação (ac, efetiv)*	Atuação necessária de um indivíduo para que um plano de ação atinja o seu objetivo. Esta atuação pode ser efetiva ou não.
Ação Adicional	*AçãoAdicional (acadic)*	Ação que faz parte de um plano de ação, mas não visa remediar uma situação, remover um erro ou ajustar uma condição.
Ação Corretiva	*AçãoCorretiva (accorr)*	Ação usada para remediar uma situação, remover um erro ou ajustar uma condição (SEI, 2002).
Critério Monitoração	*CritérioMonitoração (critmonit)*	Critério a ser utilizado na monitoração de um item de projeto a ser controlado.
Item Monitoração	*ItemMonitoração (itmonit)*	Item do projeto controlado em uma monitoração.
Monitoração	*Monitoração (monit)*	Atividade necessária à gerência do projeto, na qual vários itens do projeto são avaliados. Pode ser executada de forma planejada, em marcos ou pontos de controle, ou de forma não planejada.
Monitoração Capacitação	*MonitCapac (monitcapac)*	Item Monitoração referente a uma solicitação de capacitação feita no contexto do projeto.
Monitoração Comprometimento	*MonitCompr (monitcompr)*	Item Monitoração referente aos compromissos especificados para um dado do projeto.
Monitoração Dado Projeto	*MonitDadoProj (monitdadoproj)*	Item Monitoração referente a uma especificação de dado do projeto.

CONCEITO	PREDICADO	DEFINIÇÃO
Monitoração Envolvimento	*MonitEnvolv (monitenvolv)*	Item Monitoração referente a uma demanda de envolvimento no projeto.
Monitoração Risco	*MonitRisco (monitrisc)*	Item Monitoração referente a uma análise de risco do projeto.
Plano Ação	*PlanoAção (planac)*	Plano que propõe um conjunto de ações cuja necessidade foi percebida em uma monitoração e que se destinam a gerência pró-ativa ou reativa do projeto.

Quadro 39 – Monitoração do Projeto: Conceitos

RELAÇÃO	PREDICADO	DESCRIÇÃO
Base Monitoração Dado Projeto	*BaseMonitDadoProj (monitdadoproj, especdadoproj)*	A monitoração de dado de projeto **monitdadoproj** refere-se à especificação de dado de projeto **especdadoproj**
Base Monitoração Capacitação	*BaseMonitCapac (monitcapac, solcapac)*	A monitoração de capacitação **monitcapac** refere-se à solicitação de capacitação **solcapac**
Base Monitoração Comprometimento	*BaseMonitCompr (monitcompr, especdadoproj)*	A monitoração de compromissos **monitcompr** refere-se aos compromissos planejados na especificação de dado projeto **especdadoproj**
Base Monitoração Envolvimento	*BaseMonitEnvolv (monitenvolv, demenvolv)*	A monitoração de envolvimento **monitenvolv** refere-se à demanda de envolvimento **demenvolv**
Base Monitoração Risco	*BaseMonitRisco (monitrisc, riscproj)*	A monitoração de risco **monitrisc** refere-se ao risco projeto **riscproj**
Busca Efetividade	*AçãoAnterior (acant, acpos)*	A ação **acant** não foi efetiva e teve que ser sucedida pela ação **acpos**
	AçãoPosterior (acpos, acant)	A ação **acpos** sucedeu a ação **acant**, uma vez que a última não foi efetiva
Composição Monitoração 1	*ComposiçãoMonit1 (monit, med)*	A monitoração **monit** inclui a medição **med**
Composição Monitoração 2	*ComposiçãoMonit2 (monit, itmonit)*	A monitoração **monit** é composta pelo item de monitoração **itmonit**
Composição Plano Ação	*ComposiçãoPlanoAção (planac, ac)*	O plano de ação **planac** é composto pela ação **ac**
Motivação Item Monitoração	*MotivaçãoItemMonit (itmonit, accorr)*	O item de monitoração **itmonit** motivou a ação corretiva **accorr**
Motivação Medição	*MotivaçãoMedição (med, accorr)*	A medição **med** motivou a ação corretiva **accorr**
Participação Monitoração	*ParticipMonitoração (prof, monit)*	O profissional **prof** participou da monitoração **monit**
Produção	*Produção (monit, planac)*	A monitoração **monit** produziu o plano de ação **planac**
Utilização	*Utilização (critmonit, itmonit)*	O critério de monitoração **critmonit** foi utilizado na monitoração do item de monitoração **itmonit**

Quadro 40 – Monitoração do Projeto: Relações

NÚMERO	AXIOMA
A60	A relação de Busca Efetividade pode ser representada por dois predicados inversamente equivalentes
	$(\forall acant,acpos)\ AçãoAnterior(acant,acpos) \leftrightarrow AçãoPosterior(acpos,acant)$
A61	A relação de Busca Efetividade não é simétrica
	$(\forall acant,acpos)\ AçãoAnterior(acant,acpos) \rightarrow \neg\ AçãoAnterior(acpos,acant)$
A62	A relação de Busca Efetividade é transitiva
	$(\forall ac_1,ac_2,ac_3)\ AçãoAnterior(ac_1,ac_2) \wedge AçãoAnterior(ac_2,ac_3) \rightarrow AçãoAnterior(ac_1,ac_3)$
A63	Uma ação corretiva é motivada ou por uma medição ou por um item de monitoração
	$(\forall accorr,med,itmonit)\ (MotivaçãoItemMonit\ (itmonit,accorr) \rightarrow \neg\ MotivaçãoMedição\ (med,accorr)) \vee (MotivaçãoMedição(med,accorr) \rightarrow \neg\ MotivaçãoItemMonit\ (itmonit,accorr))$
A64	Se a relação Busca Efetividade existe, então a ação anterior não foi efetiva
	$(\forall acant,acpos)\ AçãoAnterior\ (acant,acpos) \rightarrow Ação(acant,efetiv) \wedge efetiv = N$
A65	A motivação de uma ação corretiva deve ser um componente da monitoração que produziu o plano de ação que contem a ação corretiva
	$(\forall monit_1,monit_2,accorr)\ ((\exists itmonit)\ (MotivaçãoItemMonit(itmonit,accorr) \wedge ComposiçãoMonit2(monit_1,itmonit)) \vee (\exists med)\ (MotivaçãoMedição\ (med,accorr\ ) \wedge ComposiçãoMonit1(monit_1,med))) \wedge (\exists planac)\ (ComposiçãoPlanoAção\ (planac,accorr) \wedge Produção(monit_2,planac)) \rightarrow monit_1 = monit_2$

Quadro 41 – Monitoração do Projeto: Axiomas

5.6 CONCLUSÕES

Neste capítulo, foram apresentadas as ontologias de Gerência de Requisitos e Gerência de Projetos, as quais foram elaboradas com o objetivo de: 1) auxiliar na identificação dos requisitos de informação a serem atendidos pelas ferramentas selecionadas para apoiar esses processos e, posteriormente, 2) integrar as ferramentas selecionadas, atuando como interlíngua.

CAPÍTULO 6. CONCLUSÃO

Neste capítulo, são apresentadas as considerações finais deste trabalho, as suas contribuições, bem como limitações e trabalhos futuros.

6.1 CONSIDERAÇÕES FINAIS E CONTRIBUIÇÕES

A crescente competitividade entre as empresas de desenvolvimento de software tem despertado o interesse das mesmas por certificações/avaliações que possam oferecer vantagem competitiva, ao mesmo tempo em que contribuam para a melhoria de seus processos. Ferramentas de software são essenciais para facilitar a implantação de novas práticas e facilitar a execução dos processos. Uma questão que surge é alto custo de aquisição da maioria das ferramentas.

Este trabalho tem como objetivo lançar os alicerces de uma infra-estrutura que permita compor ambientes de desenvolvimento de software baseados em ferramentas livres que atendam aos objetivos específicos da empresa. Desta forma, um processo para avaliação e seleção de ferramentas CASE foi definido e utilizado para selecionar ferramentas livres que apóiem o nível G do MPS.Br, e uma ontologia foi elaborada com o intuito de possibilitar a futura integração das ferramentas selecionadas.

As contribuições deste trabalho são:

1. **Um processo para avaliação e seleção de ferramentas**: este processo é genérico e pode ser adaptado para atender a diferentes propósitos. A utilização do processo para selecionar ferramentas livres para apoio ao nível G do MPS.Br mostra a viabilidade do mesmo, além de fornecer um exemplo de como instanciá-lo para um propósito específico;
2. **Definição dos requisitos funcionais de ferramentas que pretendam apoiar os processos do nível G do MPS.Br**: com base nos resultados esperados dos processos de Gerência de Requisitos

e Gerência de Projetos do MPS.Br, foram definidos requisitos funcionais que devem ser considerados na seleção de ferramentas que apóiem esses processos. A disponibilidade de ferramentas adequadas facilita a implementação das boas práticas prescritas nos processos;

3. **Definição das características de qualidade relevantes para ferramentas livres e para o objetivo de integração das ferramentas**: um conjunto de características de qualidade relevantes para a seleção de ferramentas livres foi definido com base nas normas ISO/IEC 9126-1 (*Engenharia de Software – Qualidade de Produto. Parte 1 – Modelo de Qualidade*) e ISO/IEC 14102 (*Information Technology – Guideline for the evaluation and selection of CASE Tools*). Cada característica proposta teve sua importância avaliada por um grupo de profissionais que trabalham com melhoria de processo, fornecendo uma baseline para a seleção de ferramentas livres. A ISO/IEC 14102 forneceu características especialmente voltadas para a integração de ferramentas;
4. **A avaliação de ferramentas livres para apoio os processos de Gerência de Projetos e Gerência de Requisitos**: várias ferramentas livres foram pesquisadas e avaliadas segundo os requisitos funcionais e não funcionais estabelecidos. Empresas que estão em busca de ferramentas para apoiar tais processos podem ser beneficiar dos resultados dessa avaliação;
5. **Uma ontologia para integração de ferramentas em conformidade com os resultados esperados no nível G do MPS.Br**: foi elaborada uma ontologia para a Gerência de Requisitos e uma para a Gerência de Projeto. O objetivo dessas ontologias é apoiar a integração das ferramentas selecionadas e, com isso, gerar um ambiente de desenvolvimento de software baseado em ferramentas livres para apoio aos processos do nível G do MPS.Br.

Durante o desenvolvimento desse trabalho, o autor orientou um trabalho de conclusão de curso de graduação (TCC) em Sistemas de Informação na Universidade Salvador (UNIFACS), no qual uma ferramenta para apoio ao processo de avaliação e seleção de ferramentas, chamada *Sistema Chooser*, foi desenvolvida. Essa ferramenta encontra-se em estado operacional e pode ser obtida através de contato com o autor deste trabalho.

6.2 LIMITAÇÕES

As limitações deste trabalho consistem em:

1. **Falta de avaliação das ontologias:** as ontologias não foram codificadas em um ambiente que permitisse a verificacão da correcão dos seus axiomas e nem foram avaliadas formalmente por um grupo de especialistas no domínio. A codificação e a avaliação formal são necessárias para, respectivamente, assegurar a consistência e a adequação da ontologia.
2. **Defasamento das ferramentas pesquisadas:** as ferramentas foram pesquisadas em 2006 e a avaliação foi feita no inicio de 2007. A partir daí, podem ter surgido ferramentas mais robustas que atendam mais adequadamente aos propósitos estabelecidos. O ideal seria refazer a pesquisa de tempos em tempos.
3. **Subjetividade dos critérios para a classificação dos requisitos funcionais e para escolha das métricas para os requisitos não funcionais:** o conjunto de ferramentas selecionadas ao final do processo pode ser afetado pelos critérios adotados para a classificação dos requisitos funcionais em requisitos críticos e pelos critérios utilzados para determinação dos pesos associados a esses requisitos. Da mesma forma, a escolha das métricas para avaliação dos requisitos não funcionais pode afetar o resultado final do processo. Empresas que queiram utilizar os resultados da avaliação

realizada neste trabalho devem verificar a adequação desses critérios para o seu contexto.

6.3 TRABALHOS FUTUROS

Como trabalhos futuros, pretende-se:

1. Avaliar as as ontologias de Gerência de Requisitos e de Gerência de Projeto;
2. Gerar as ontologias em OWL, a fim de facilitar a integração das ferramentas;
3. Realizar a integração das ferramentas selecionadas, a fim de prover o ambiente de desenvolvimento de software para apoio ao nível G do MPS.Br.

REFERÊNCIAS

ALMEIDA, Mauricio B.; BAX, Marcello P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. **Ciência de Informação,** Brasília, p. 7-20. set. 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12207**: tecnologia de informação – processos de ciclo de vida do software.Rio de Janeiro, 1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1**: engenharia de software – qualidade de produto. Parte 1 – modelo de qualidade. Rio de Janeiro, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-2**: engenharia de software – qualidade de produto. Parte 2 – métricas externas. Rio de Janeiro, 2002.

BERTOLLO, Gleidson, et al. ODE – Um ambiente de desenvolvimento de software baseado em ontologias. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 16., 2002, Gramado, **Anais...** Rio Grande do Sul: Centro de Treinamento e Eventos, 2002, p. 438-443. Disponível em: <http://www.inf.ufes.br/~falbo/download/pub/CFSbes2002.pdf>. Acesso em: 01 ago. 2005.

BERTOLLO, Gleidson, FALBO, Ricardo. Establishing a common vocabulary for software organizations understand software processes. In: EDOC INTERNATIONAL WORKSHOP ON VOCABULARIES, ONTOLOGIES AND RULES FOR THE ENTERPRISE, 9., 2005, Enschede, **Proceedings…** Netherlands: University of Twente, 2005.

CHANDRASEKARAN, Balakrishnan; JOSEPHSON, John R.; BENJAMINS, Richard. What are ontologies, and why do we need them? **Intelligent systems and their applications**, IEEE, v. 14, n. 1, p.20-25, jan. 1999.

DIAS, Márcio G. **Organização do conhecimento utilizado na manutenção de software.** 2003. 220 f. Dissertação (Mestrado) - Curso de Gestão do Conhecimento e da Tecnologia da Informação, Departamento de Ciência da Computação, Universidade Católica de Brasília, Brasíia, 2003.

FALBO, Ricardo. **Integração de conhecimento em um ambiente de desenvolvimento de software.** 1998. 217 f. Tese (Doutorado) - Curso de Ciências em Engenharia de Sistemas e Computação – Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, 1998.

FALBO, Ricardo; DUARTE, Katia C. Uma ontologia de qualidade de software. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 16., 2000, João Pessoa, **Anais ...** Paraiba: Hotel Tambaú, 2000, p. 275-285.

FALBO, Ricardo; MIAN, Paula G.; NATALI, Ana C. Ambientes de desenvolvimento de software e o projeto ADS. **Engenharia, Ciência e Tecnologia**, Vitória-ES, v. 4, n. 4, p.3-10, 10 jul. 2001.

FÉRNANDEZ, Mariano; GÓMEZ-PÉREZ, Assunción; JURISTO, Natalia. Methontology: from ontological art towards ontological engineering. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELIGENCE, 16., 1997, Stanford, **Proceedings …** California, 1997

FIKES, Richard; FARQUHAR, Adam. Distributed repositories of highly expressive reusable ontologies. **IEEE Intelligent Systems & their applications**, IEEE, v. 14, n. 2, p. 73-79, Mar. 1999.

FOX, Mark S., CHIONGLO, John F., FADEL, Fadi G. A common-sense model of the enterprise. In: INTERNATIONAL INDUSTRIAL ENGINEERING CONFERENCE PROCEEDINGS, 1., 1993, Los Angeles, **Proceedings…** California, 1993.

FUGGETTA, Alfonso. Software process: a roadmap. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 20., 2000, Limerick, **Proceedings…** Ireland, 2000.

GENESERETH, Michael R. **Knowledge interchange format**. Disponível em: <http://logic.stanford.edu/kif/dpans.html>. Acesso em: 6 fev. 2009.

GRUBER, Thomas R. Ontolingua: A mechanism to support portable ontologies - version 3.0, **Knowledge Systems Laboratory - Stanford University**, Jun. 1992. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.9819>. Acesso em: 2 out. 2009.

GRUBER, Thomas R. Towards principles for the design of ontologies used for knowledge sharing. **International Journal of Human-Computer Studies**, San Diego, p. 907-928. 20 ago. 1995.

GRUBER, Thomas R., **What is an Ontology**. Disponível em: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>. Acesso em: 16 jul. 2007

GRÜNINGER, Michael; FOX, Mark S. Methodology for the design and evaluation of ontologies. In: WORKSHOP ON BASIC ONTOLOGICAL ISSUES IN KNOWLEDGE SHARING, 1995, Montreal, **Proceedings** Canadá, 1995.

GUARINO, Nicholas. Formal ontology, conceptual analysis and knowledge representation. **International Journal of Human-Computer Studies**, San Diego, p. 625-640, 20 ago., 1995.

GUARINO, Nicholas. Formal ontology and information systems. In: FORMAL ONTOLOGY IN INFORMATION SYSTEMS, 1., 1998, Trento, **Proceedings…** Italy, 1998.

GUIMARÃES Fernando J. Z. **Utilização de ontologias no domínio B2C**. 2002. 195f. Dissertação (Mestrado) – Curso de Ciência da Computação, Departamento de Informática, Pontifícia Universidade Católica - PUC, Rio de Janeiro, 2002.

HEXSEL, Roberto A. **Software livre:** propostas e ações do governo para incentivar o uso do software livre. Paraná: Universidade Federal do Paraná (UFPR), 2002, 53f.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / INTERNATIONAL ELECTRONICHAL COMMISION. **ISO/IEC 14102.** Information technology – guideline for the evaluation and selection of CASE Tools. 1995.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / INTERNATIONAL ELECTRONICHAL COMMISION. **ISO/IEC 14598-5**. Information technology – software product evaluation – part 5: process for evaluators. 1998**.**

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / INTERNATIONAL ELECTRONICHAL COMMISION. **ISO/IEC 25001**. Software engineering – software product quality requirements and evaluation (SquaRE) – Guide to SquaRE. 2005.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION / INTERNATIONAL ELECTRONICHAL COMMISION. **ISO/IEC 90003:2004**. Software engineering – guidelines for the application of ISO 9001:2000 to computer software. 2004.

J. FILHO, Fernando. **Projeto arquitetural**. São Paulo: Universidade de Campinas, 2003. 10f.

KITCHENHAM, Barbara, *et al.* Towards an ontology of software maintenance, **Journal of software maintenance: research and practice**, California, p. 365-389, 21 dez. 1999.

MELHORIA DE PROCESSO DE SOFTWARE BRASILEIRO - MPS.Br. **Guia geral, versão 1.2**. Softex, 2007. 52.p. Disponível em: http://www.softex.br/mpsbr/_guias/ default.asp Acesso em: 10 mar. 2009.

MELHORIA DE PROCESSO DE SOFTWARE BRASILEIRO - MPS.Br. **Avaliações MA-MPS**. Relatório Técnico de Avaliações MPS.Br. Disponível em: <http://www.softex.br/mpsbr/avaliacoes/ default.asp>. Acesso em 10 mar. 2009.

MIAN, Paula G. **ODE:** uma ferramenta de apoio ao desenvolvimento de ontologias. 2003. 191f. Dissertação (Mestrado) – Curso de Ciência da Computação, Universidade Federal do Espírito Santo – UFES, Espírito Santo, 2003.

OLIVEIRA, Kathia M.. **Modelo para construção de ambientes de desenvolvimento de software orientados a domínio**. 1999. 220f. Tese (Doutorado) – Curso de Ciências em Engenharia de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, 1999.

PRESSMAN, Roger S. **Software engineering :** a practioner's approach. 5.ed. New York: McGraw-Hill, 2001.

REED, Karl. Software engineering – a new millenium?. **IEEE Software**, IEEE, v. 17, n. 4, p. 107, Jul. 2000.

ROCHA, Ana R.; MALDONADO; José C.; WEBER, Kival C. **Qualidade de software:** teoria e prática. São Paulo: Prentice Hall, 2001.

RUSSELL, Stuart; NORVIG, Peter. **Artificial intelligence :** a modern approach. New York: Prentice Hall Series in Artficial Intelligence ,1995.

SANTIN, Carlos E., et al. Abordagem de representação de conhecimento visual em três níveis. In: SEMINÁRIO DE PESQUISA EM ONTOLOGIA, 1., 2008, Rio de Janeiro, **Trabalhos Aprovados...**, Rio de Janeiro: Departamento de Ciência da Informação. Disponível em: <http://www.uff.br/ontologia/artigos/322.pdf>. Acesso em 02 out. 2009.

SOFTWARE ENGINEERING INSTITUTE. **CMMI:** capability maturity model integration. Version 1.1 - Staged Representation, Pittsburgh, 2002. 639.p

SOFTWARE ENGINEERING INSTITUTE. **CMMI:** capability maturity model integration for development. Version 1.2, Pittsburgh, 2006. 573.p

SOFTWARE LIVRE. **Secretário fala sobre software livre**. Disponível em: <http://www.softwarelivre.gov.br/noticias/News_Item.2006-06-06.4923>. Acesso em: 31 jan. 2007.

SOMMERVILLE, Ian. **Engenharia de software**. 6.ed. São Paulo: Pearson Education, 2004.

SWEBOOK. **Guide to software engineering body of knowledgment**. Los Alamitos – California: IEEE Computer Society, 2004.

USCHOLD, Mike; GRUNINGER, Michael. ONTOLOGIES: principles, methods and applications. **Journal of knowledge engineering review**, Cambridge, p. 93-136, jun. 1996.

VALENTE, André. **Legal knowledge engineering:** a modelling approach. Amsterdã: Universidade de Amsterdã, 1995. 224f.

VAN HEIJST, GertJan. **The role of ontologies in knowledgment engineering**, 1995. 243f. Tese (Doutorado) Curso de Ciência da Computação, Universidade de Amsterdã, Amsterdã, Holanda, 1995.

VAN HEIJST, GertJan; VAN SCHREIBER, Antony; WIELINGA, Bob, Using explicit ontologies in KBS development, **International Journal of human-computer studies**, San Diego, p. 183-292, 17 jul. 1997.

VILLELA, Karina. **Definição e construção de ambientes de desenvolvimento de software orientados a organização**. 2004. 330 f. Tese (Doutorado) – Curso de Engenharia de Sistemas e Computação – Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, 2004.

VILLELA, Karina. Projeções tecnológicas em qualidade de software. In: SEMANA DA CIÊNCIA E TECNOLOGIA, 1., 2006, Salvador, **Estudos prospectivos sobre tendências em TI...** Bahia, 2006.

W3C. **Web Ontology Language**. Disponível em: <http://www.w3.org/TR/owl-features/>. Acesso em 04 ago. 2009.

Apêndice A – Resultado da Aplicação do Questionário para Obtenção de Pesos dos Requisitos Não Funcionais

A.1 Questionário Aplicado

Características obtidas das normas 9126-1 (Engenharia de Software - Qualidade de Produto. Parte 1 - Modelo de Qualidade) e 14102 (Tecnologia de Informação - Guia para Avaliação e Seleção de Ferramentas CASE)

Caso você estivesse adquirindo uma ferramenta livre para apoio à implantação das práticas do CMMI, qual peso você atribuiria para cada uma das características a seguir?

Preencha a coluna de Peso, considerando a escala abaixo:

0. Sem Importância
1. Pouca Importância
2. Importante
3. Muito Importante
4. Imprescindível

Funcionalidade: existência de um conjunto de funções que atendam às necessidades explícitas ou implícitas do usuário, quando a ferramenta estiver sendo utilizada sob as condições especificadas		
Sub-Característica	**Definição**	**Peso**
Adequação	Capacidade da ferramenta prover um conjunto apropriado de funções para os objetivos e tarefas que foram especificados pelo usuário	
Acurácia (Exatidão)	Capacidade da ferramenta prover resultados ou efeitos corretos ou conforme acordados	
Segurança de Acesso	Capacidade da ferramenta proteger informações e dados	
Interoperabilidade [1]: capacidade da ferramenta de interagir com uma ou mais ferramentas especificadas		
Sub-Característica	**Definição**	**Peso**
Integração de Dados	Capacidade da ferramenta de usar, processar e distribuir informações compartilhadas com um conjunto de ferramentas	
Integração de Interface	Capacidade da ferramenta de apresentar uma interface consistente com um conjunto de ferramentas	
Integração de Controle	Capacidade da ferramenta em interagir com um conjunto de ferramentas ou ambientes de software	
Integração de Conhecimento	Capacidade da ferramenta de disponibilizar o acesso aos seus metadados	

Confiabilidade: capacidade da ferramenta de manter um nível de desempenho especificado, quando usada sob as condições especificadas		
Sub-Característica	**Definição**	**Peso**
Maturidade	Capacidade da ferramenta evitar falhas decorrentes de defeitos na ferramenta	
Tolerância a Falhas	Capacidade da ferramenta manter um nível de desempenho especificado em casos de falha da ferramenta ou de violação da interface especificada.	
Recuperabilidade	Capacidade da ferramenta de re-estabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha	
Utilização de Rotinas de Backup	Capacidade da ferramenta de prover e utilizar backup	
Usabilidade: capacidade da ferramenta de ser compreendida, aprendida e operada de forma atraente ao usuário, quando usada sob as condições especificadas		
Sub-Característica	**Definição**	**Peso**
Inteligibilidade	Capacidade da ferramenta de possibilitar ao usuário compreender que ela é apropriada para as tarefas e condições de uso especificadas	
Apreensibilidade	Capacidade da ferramenta de possibilitar ao usuário aprender a sua utilização	
Operacionalidade	Capacidade da ferramenta de permitir que o usuário a opere facilmente	
Atratividade	Capacidade da ferramenta de ser atraente ao usuário	
Eficiência: capacidade da ferramenta de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob as condições especificadas		
Sub-Característica	**Definição**	**Peso**
Comportamento em relação ao Tempo	Capacidade da ferramenta de oferecer taxa de transferência e tempo de resposta e de processamento apropriados, quando executada sob as condições especificadas	
Utilização de Recursos	Capacidade da ferramenta de usar o tipo e quantidade de recursos apropriados, quando executada sob as condições especificadas	
Manutenibilidade: capacidade da ferramenta de ser modificada. As modificações podem incluir correções, melhorias ou adaptações		
Sub-Característica	**Definição**	**Peso**
Analisabilidade	Capacidade da ferramenta de permitir o diagnóstico de defeitos ou causas de falhas, e/ou identificação de partes a serem modificadas	
Modificabilidade	Capacidade da ferramenta de permitir que uma modificação especificada seja implementada	
Estabilidade	Capacidade da ferramenta de evitar efeitos inesperados decorrentes de modificações	
Testabilidade	Capacidade da ferramenta de permitir que ela seja validada, quando modificada	
Frequência de Atualizações [2]	Política da organização de fornecer atualizações regulares para corrigir problemas ou incluir funcionalidades	

Portabilidade: capacidade da ferramenta de ser transferida de um ambiente para outro		
Sub-Característica	**Definição**	**Peso**
Adaptabilidade	Capacidade da ferramenta de ser adaptada para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pela ferramenta	
Capacidade para ser instalado	Capacidade da ferramenta de ser instalada em um ambiente especificado	
Coexistência	Capacidade da ferramenta de coexistir com outros produtos de software independentes, em um ambiente comum, compartilhando recursos	
Capacidade para substituir	Capacidade da ferramenta de ser usada em substituição a outro produto de software especificado, com o mesmo propósito e no mesmo ambiente	

[1] Esta característica é uma das sub-características da característica Funcionalidade na norma 9126-1 (Engenharia de Software - Qualidade de Produto. Parte 1 - Modelo de Qualidade). Porém, na norma 14102 (Tecnologia de Informação - Guia para Avaliação e Seleção de Ferramentas CASE), ela aparece como característica, contendo outras sub-características. Diante do objetivo do projeto, que é integracao de ferramentas, resolvemos levar em consideração a segunda norma, o que permitirá avaliar as sub-características relacionadas à Interoperabilidade.

[2] Esta sub-característica não é mencionada nas normas consultadas, mas foi incluída por a considerarmos importante para a seleção de ferramentas livres

A.2 Questionário com Resultados Tabulados

Funcionalidade: existência de um conjunto de funções que atendam às necessidades explícitas ou implícitas do usuário, quando a ferramenta estiver sendo utilizada sob as condições especificadas

Sub-Característica	Número do Avaliador												TOTAL SEM PESOS	AVALIADORES PESO 1		AVALIADORES PESO 2		TOTAL COM PESOS
	1	2	3	4	5	6	7	8	9	10	11	12		TOTAL	6	TOTAL	6	
Adequação	2	4	4	3	3	4	2	4	3	3	4	4	3,3	3,3		3,3		3,3
Acurácia (Exatidão)	4	4	4	4	4	4	2	4	4	4	4	4	3,8	4		3,7		3,8
Segurança de Acesso	2	3	3	3	2	4	4	2	4	2	2	3	2,8	2,8		2,8		2,8

Interoperabilidade: capacidade da ferramenta de interagir com uma ou mais ferramentas especificadas

Sub-Característica	Número do Avaliador												TOTAL SEM PESOS	AVALIADORES PESO 1		AVALIADORES PESO 2		TOTAL COM PESOS
	1	2	3	4	5	6	7	8	9	10	11	12		TOTAL	6	TOTAL	6	
Integração de Dados	3	2	3	3	3	3	3	4	2	3	4	4	3,1	2,8		3,3		3,2
Integração de Interface	1	2	2	3	1	3	2	2	1	0	2	3	1,8	2		1,7		1,8
Integração de Controle	2	2	3	2	2	3	3	2	2	1	3	3	2,3	2,3		2,3		2,3
Integração de Conhecimento	1	3	2	3	1	2	2	1	3	2	4	3	2,3	2		2,5		2,3

Confiabilidade: capacidade da ferramenta de manter um nível de desempenho especificado, quando usada sob as condições especificadas																		
Sub-Característica	**Avaliadores**												**TOTAL SEM PESOS**	**AVALIADORES PESO 1**		**AVALIADORES PESO 2**		**TOTAL COM PESOS**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**TOTAL**	**6**	**TOTAL**	**6**	
Maturidade	3	4	3	4	4	3	3	2	4	3	4	2	3,3	3,5		3		3,2
Tolerância a Falhas	3	4	4	3	4	3	2	2	3	3	2	1	2,8	3,5		2,2		2,6
Recuperabilidade	4	4	4	4	4	2	4	2	4	2	4	1	3,3	3,7		2,8		3,1
Utilização de Rotinas de Backup	3	4	4	2	4	4	4	1	4	4	2	1	3,1	3,5		2,7		2,9

Usabilidade: capacidade da ferramenta de ser compreendida, aprendida e operada de forma atraente ao usuário, quando usada sob as condições especificadas																		
Sub-Característica	**Avaliadores**												**TOTAL SEM PESOS**	**AVALIADORES PESO 1**		**AVALIADORES PESO 2**		**TOTAL COM PESOS**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**TOTAL**	**6**	**TOTAL**	**6**	
Inteligibilidade	2	4	3	3	3	3	3	2	3	4	4	3	3,1	3		3,2		3,1
Apreensibilidade	2	3	3	2	3	3	4	1	2	3	3	3	2,7	2,7		2,7		2,7
Operacionalidade	2	4	3	2	3	3	4	2	3	3	3	3	2,9	2,8		3		2,9
Atratividade	2	3	2	1	3	3	3	2	0	1	2	2	2	2,3		1,7		1,9

Eficiência: capacidade da ferramenta de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob as condições especificadas																		
Sub-Característica	**Avaliadores**												**TOTAL SEM PESOS**	**AVALIADORES PESO 1**		**AVALIADORES PESO 2**		**TOTAL COM PESOS**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**TOTAL**	**6**	**TOTAL**	**6**	
Comportamento relação ao Tempo	2	4	3	0	3	3	4	2	4	2	4	4	2,9	2,5		3,3		3,1
Utilização de Recursos	3	4	2	0	2	3	3	2	4	2	2	4	2,6	2,3		2,8		2,7

Manutenibilidade: capacidade da ferramenta de ser modificada. As modificações podem incluir correções, melhorias ou adaptações																		
Sub-Característica	**Avaliadores**												**TOTAL SEM PESOS**	**AVALIADORES PESO 1**		**AVALIADORES PESO 2**		**TOTAL COM PESOS**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**TOTAL**	**6**	**TOTAL**	**6**	
Analisabilidade	2	3	3	2	4	3	3	2	4	1	3	3	2,8	2,8		2,7		2,7
Modificabilidade	2	3	3	3	3	3	3	2	4	3	3	4	3	2,8		3,2		3,1
Estabilidade	2	3	3	4	2	3	3	2	4	4	3	3	3	2,8		3,2		3,1
Testabilidade	2	3	3	3	2	3	4	2	3	3	3	3	2,8	2,7		3		2,9
Frequência de Atualizações [2]	2	4	3	0	4	2	3	2	4	3	3	2	2,7	2,5		2,8		2,7

Portabilidade: capacidade da ferramenta de ser transferida de um ambiente para outro																		
Sub-Característica	**Avaliadores**												**TOTAL SEM PESOS**	**AVALIADORES PESO 1**		**AVALIADORES PESO 2**		**TOTAL COM PESOS**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**		**TOTAL**	**6**	**TOTAL**	**6**	
Adaptabilidade	3	3	3	3	1	3	4	1	1	2	2	4	2,5	2,7		2,3		2,4
Capacidade para ser instalado	3	4	4	3	4	2	4	2	3	3	3	4	3,3	3,3		3,2		3,2
Coexistência	3	4	2	4	3	3	4	2	4	4	3	4	3,3	3,2		3,5		3,4
Capacidade para substituir	3	3	2	2	3	2	3	2	3	1	2	3	2,4	2,5		2,3		2,4

Apêndice B – Ferramentas Pesquisadas para as Áreas de Processo do Nível G do MPS.Br

As ferramentas em destaque (com fundo cinza) foram as ferramentas pré-selecionadas.

FERRAMENTA	URL	PLATAFORMA	DESCRIÇÃO
Controla	http://baixaki.ig.com.br/site/detail34971.htm	Plataforma Windows	O Controla é uma ferramenta CASE para apoio ao processo de desenvolvimento de software que permite ao analista ou ao gerente de projetos efetuar os controles necessários durante o processo de desenvolvimento de um produto de software.
OSRMT	http://www.osrmt.com/	Independente de SO	Requirements management tool designed to achieve full SDLC traceability for features, requirements, design, implementation and testing. UI for requirements derivation, version control, common or custom attributes - rationale, source, risk, effort etc
RTH	http://www.rth-is-quality.com/home.php	Independente de SO	RTH is a web-based testing framework. RTH handles requirement and test management, stores manual and automated test results and has a built in bugtracker. Also includes reporting (requirement coverage, testing status, defects found during test, etc).
atREQID	http://sourceforge.net/projects/atreqid	Não informado	This project aims at providing a set of tools for gathering, validating, maintaining and publishing requirements. Formulating requirements and verifying that requirements are met is a key for the success of a software engineering project
CMF Requirements Manager	http://sourceforge.net/projects/cmfreq	Independente de SO	CMFReq is a project to manage lists of requirements for managing software projects. It is a CMF project, but is primarily intended to be included in a plone site
*CRM-CTT - Online collaboration**	http://sourceforge.net/projects/crm-ctt	Independente de SO	CRM-CTT is to be used for tracking tickets, requirements, complaints, etc. It is suitable for any department in which something comes in, must be handled and goes out using workflows. It creates Excel sheets, invoices, custom reports, and more
DRES - Requirements Engineering Support	http://sourceforge.net/projects/dres	Não informado	First free and open-source requirements management software! PHP application that uses MySQL database or CORBA server to store and manage your project's requirements. Rich attributes set, extensive XML support, hierarchical folders, reports.
JEREMIA - Requirement management system	http://sourceforge.net/projects/jeremia	Independente de SO	Jeremia is an open source requirement management system. With Jeremia it is possible to describe requirements and requirement documentation within one system. The requirements and requirement documents are stored within a database
Magna Requirements	http://sourceforge.net/projects/magnadev	Não informado	This project seeks the creation of a Requirements Development and Management program for analysts and developers
NEORMS	http://sourceforge.net/projects/neorms	Não disponível	NEORMS -- Net-Enabled Open Requirements Management System A tool for maintaining and tracking requirements over the project lifecycle

FERRAMENTA	URL	PLATAFORMA	DESCRIÇÃO
OpenRequirements	http://sourceforge.net/projects/openreqs	Independente de SO	Generally useful requirements management system, initially defined for software development.
Plandora Project Management	http://sourceforge.net/projects/plandora	Independente de SO	Requirements or Tickets (Incidents) control; Tasks planning using a Gantt Chart; Store the 'project history' (Tasks and Requests); Follow the Balanced Scorecard of Projects (customized)
ReqBook	http://sourceforge.net/projects/reqbook	Independente de SO	A console/GUI tool for software requirements management. The feature list will include usable rich text requirements editing, HTML/PDF/WordML output, requirements IDs and groups support, CVS intergration, automatic changesets generation, etc
REQ-dot-NET	http://sourceforge.net/projects/req-dot-net	Independente de SO	REQ-dot-NET is a requirements management solution based around KISS. It's a low-cost implementation environment looking to help projects gather, quantify and fulfill requirement analyst's data and further be an asset to the development team.
reqs	http://reqs.tigris.org/	Independente de SO	
ReqSimile	http://sourceforge.net/projects/reqsimile	Independente de SO	ReqSimile is an application that supports and facilitates linking and merging between requirements specifications written in plain text. The underlying similarity calculations are based on the latest research in the field.
Require Freedom Free requirements tool	http://sourceforge.net/projects/requirefreedom	Não disponível	A GPL tool for system requirements that allows: 1) Data entry of requirements 2) Version control of requirment 3) Read access to requirments via HTTP and/or other open protocols for use in other tools and documents (e.g. Specification documents, test
Requirements Automation and Validation Engine	http://rave.tigris.org/	Independente de SO	
Requirements Manager	http://sourceforge.net/projects/requirements	Independente de SO	Software for tracking requirements, business rules, implementations of the requirements and resources associated to each item. Manages the responsibles and with alert system it informs when a change has occured and forces revision of other artifacts.
Requirements Manager	http://sourceforge.net/projects/reqman	Linux	This project provides a tool for requirement capture, usecase capture and requirement analysis in a multiuser setup
Requirements Manager	http://sourceforge.net/projects/reqsman	Independente de SO	A tool for Requirements Engineering, Management, and Development. Possible features: process template support, stakeholder management, prototype integrations, use cases, requirements reuse, source-code & design dependencies, surveys, and traceability.
Requirements Tracker	http://sourceforge.net/projects/reqtracker	Não informado	A system for recording, classifiying, prioritizing, and organizing requirements for software development projects. It is based on best practices for requirements engineering from Requirements Engineering: A good practice guide (Sommerville & Sawyer,1997)

FERRAMENTA	URL	PLATAFORMA	DESCRIÇÃO
Requirements Tracker	http://sourceforge.net/projects/reqtrack	Independente de SO	A web based project management tool that uses Requirements Tracking as its central paradigm. Uses Tomcat as the application server and will be highly modular.
Requirements Tracker Tool	http://sourceforge.net/projects/myrequirement	Independente de SO	Provide a requirement track system during software development. 1. requirements are split into detail features/packages 2. the discusson (i.e. email/snapshot) could be connected to the requirement 3. progress and plan are recorded
Reqzilla requirements management system	http://sourceforge.net/projects/reqzilla	Linux	Reqzilla is an open-source, web-based requirements management system
RSQSim	http://sourceforge.net/projects/softprosim	Não informado	The project is aimed at producing software for simulating Software Development Processes and in particular Requirements Engineering. Written in Java using DSO
Scarab	http://scarab.tigris.org/	Não informado	Scarab is a highly customizable artifact tracking system
RT	http://www.bestpractical.com/rt	Independente de SO	
SRM-System Requirements Manager	http://sourceforge.net/projects/srmbraga	Independente de SO	SRM (System Requirements Manager)- This project have the purpose to create a tool for manage requirements.
TauReq	http://sourceforge.net/projects/taureq	Independente de SO	Requirements management tool
TrackIt	http://sourceforge.net/projects/trackit	Independente de SO	TrackIt - The Dev Portal. Technical Management of software projects featuring Tickets, Requirements, Features, CVS, Subversion, Nightly Build, Wiki, News, user customizable content, and an extensible architecture for future functionality
usecase	http://sourceforge.net/projects/usecase	Independente de SO	UseCase is web application for people who care about requirements. It provides a userfriendly interface to communicate and collect the needs of the users
WishList	http://sourceforge.net/projects/reqwishlist	OS X	WishList is a requirements management tool which allows requirements management on a generic level named "Domain" and a platform specific level "Platform" which defines project templates for the "Project" level.
Não foram encontradas ferramentas no site http://www.savannah.nongnu.org			

Tabela B.1 – Ferramentas de Gerência de Requisitos

FERRAMENTA	URL	PLATAFORMA	DESCRIÇÃO
DOTPROJECT	http://sourceforge.net/projects/dotproject/	Independente de SO	PHP web-based project management framework that includes modules for companies, projects, tasks (with Gantt charts), forums, files, calendar, contacts, tickets/helpdesk, multi-language support, user/module permissions and themes
Gantt Project Tool	http://ghannid.homeip.net/wiki/LGanttPrd_en.wiki	Não disponível	lGantt © is a Java Language Application used to create Gantt Charts and help in evaluating time based activities. This application has been in development since november of 2004 and currently is an alpha release
NetOffice	http://sourceforge.net/projects/netoffice/	Independente de SO	
Open WorkBench	http://www.openworkbench.org/	Windows 2000, Server 2003, XP	Open Workbench is an open source desktop application that provides robust project scheduling and management functionality. Already the scheduling standard for more than 100,000 project managers worldwide, Open Workbench is a free and powerful alternative to Microsoft Project.
PHProjekt	http://www.phprojekt.com/	Não disponível	PHProjekt is a modular application for the coordination of group activities and to share informations and document via intranet and internet. Components of PHProjekt: Group calendar, project management, time card system, file management, contact manager, mail client and 9 other modules
Streber	http://www.streber-pm.org/	Não disponível	With Streber Freelancers and small teams can easily setup projects and keep track of tasks, issues, bugs, efforts etc. Project-User-Rights can be adjusted e.g. to provide clients a limited view of the current project state.
EZEstimate	http://www.openrage.com/main/ezestimate_full.htm	Windows 95/98/2000/XP	
Construx	www.construx.com/resources/estimate	Windows	
SEAT (Software Estimation and Analysis Tool)	http://www.cs.umt.edu/RTSL/dsstud/dsstudio.htm	Windows	
Austin	http://austin.sourceforge.net/	Windows	Austin is a program for project management. It is a command line Windows program that manipulates project timelines in XML. It can produce HTML output and Gantt charts in PNG format
Achievo	http://www.achievo.org/	Independente de SO	Achievo is a flexible web-based free project management tool for small to medium businesses. Achievo's project management capabilities will enable organizations to support their business processes in a simple, but effective manner

FERRAMENTA	URL	PLATAFORMA	DESCRIÇÃO
Angel Project	http://dec.bmth.ac.uk/ESERG/ANGEL/	Independente de SO	
Double Choco Late	http://sourceforge.net/projects/dcl	Independente de SO	Double Choco Latte is a project to create a solution for managing some IT departments including software development and call center activity. It has a web interface and will also have a stand-alone Java client
GanttPV	http://www.pureviolet.net/ganttpv/	Não disponível	Many project managers will appreciate a simple, open-source tool that will help them to manage their projects.
GNUe	http://www.gnu.org/software/gnue/project/what.html	Não disponível	GNU Enterprise is an ERP where project management is included
IPM	http://sourceforge.net/projects/ipm	Independente de SO	IPM makes project tracking easier. Complete with project-specific task lists, and statistics. The interface is customizable, and easy to use. This is simple web-based project tracking at its best
JProjectTimer	http://www.h.shuttle.de/mitch/jprojecttimer.en.html	Independente de SO	JProjectTimer helps you to plan a project timeline. You can enter tasks, their duration, and their dependencies relative to other tasks. Afterwards, you can also print out a Gantt-diagram which gives an overview of your project
Kplato	http://www.koffice.org/kplato/	Não disponível	KPlato is a project management application. In this first public release we focus on planning and scheduling of projects
LEAP	http://csdl.ics.hawaii.edu/Tools/LEAP	Independente de SO	
Maven	http://maven.apache.org/	Não disponível	Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.
NetOffice Dwins	http://netoffice.dwins.com/	Não disponível	netOffice Dwins is a free web based time tracking, timesheet, and project management environment. Use netOffice Dwins for employee timesheets, time tracking, and project management. netOffice Dwins allows managing and sharing information about teams, projects, tasks, deadlines. It also allows anyone with an internet connection to enter time and submit it for approval. Approval requests can be routed to managers, project managers, and HR members electronically.
OpenOffice.org Project Management Tool Project	http://oopm.openoffice.org/	Não disponível	A project in its initial stage but is very promising
Opensched	http://sourceforge.net/projects/opensched	Linux/UNIX	This program is used to automatically schedule resources for a project. For instance, if 5 programmers cooperate on a software development project, and 100 tasks must be done, opensched can capture information about who can do what, what tasks depend

FERRAMENTA	URL	PLATAFORMA	DESCRIÇÃO
PHPCollab	http://sourceforge.net/projects/phpcollab/	Independente de SO	Project management and collaboration over the internet. Features team/client sites, task assignment, document repository/workflow, gantt charts, discussions, calendar, notifications, support requests, weblog newsdesk, invoicing, and many other tools
Planner	http://developer.imendio.com/wiki/Planner	Linux/UNIX	Planner is a project managment tool for the Gnome desktop, for planning, scheduling and tracking projects. The project is run by Imendio, who develops the software in cooperation with the Planner community
Project/Open	http://www.project-open.org	Independente de SO	
Project Planner	http://www.brothersoft.com/Business_Project_Management_Project_PlannerPE_12724.html	Windows 95/98/ME/2000	
SimmanTools	http://simmantools.sourceforge.net/	Linux/UNIX	Simple Management Tools is a set of web based, simple, commonly used management tools
TaskJuggler	http://www.taskjuggler.org/	Não disponível	TaskJuggler is a modern and powerful project management tool. Its new approach to project planning and tracking is far superior to the commonly used Gantt chart editing tools. It has already been successfully used in many projects and scales easily to projects with hundreds of resources and thousands of tasks
TUTOS	http://www.tutos.org/homepage/index.html	Não disponível	TUTOS is a tool to manage the organizational needs of small groups, teams, departments ...
USC COCOMO II	http://sunset.usc.edu/research/COCOMOII/index.html	Windows	
XPLANNER	http://www.xplanner.org/	Independente de SO	XPlanner is a project planning and tracking tool for eXtreme Programming (XP) teams. f the team appears to be overcommitted, the set of stories are renegotiated with the customer. The XPlanner tool was created to support this process and address issues experienced in a long-term real-life XP project
Veritime Tracker	http://www.brothersoft.com/Business_Project_Management_VeriTime_Time_Tracker_15485.html	Windows 95/98/2000/XP	
Zentrack	http://www.zentrack.net/	Não disponível	Zentrack is a fully customizable help desk, bug tracking, and project management system. It is designed for systems with less than 10,000 users (until the 3.0 release!)

Tabela B.2 – Ferramentas de Gerência de Projetos

Apêndice C – Modelo de Descrição de Ferramentas

CARACTERIZAÇÃO DE FERRAMENTAS

Nome:	Escrever o nome da ferramenta
Apelido	Escrever o nome que será referenciado na avaliação da ferramenta
URL:	Escrever a URL em que a ferramenta pode ser obtida
Área(s) de Processo:	Escrever a(s) área(s) à(s) qual(is) se destina a ferramenta
Instituição ou Pessoa Mantenedora:	ex.:Sourceforge.net
Base de Usuários:	Escrever usuários que já utilizam essa ferramenta
Objetivo Declarado:	Escrever o objetivo declarado da ferramenta
Requisito(s) de HW:	Escrever os requisitos de HW necessários para conseguir executar a ferramenta
Requisito(s) de SW:	Escrever os requisitos de SW necessários para conseguir executar a ferramenta
Sistema(s) Operacional(is):	Escrever o(s) Sistema(s) Operacional(is) nos quais a ferramenta executa
Principal(is) Característica(s):	
Característica 1:	Descrição da característica 1
Característica 1:	Descrição da característica 2
Característica 3:	Descrição da característica 3
Característica 4:	Descrição da característica 4
Característica 5:	Descrição da característica 5
Característica 6:	Descrição da característica 6
Característica 7:	Descrição da característica 7
Característica 8:	Descrição da característica 8
Característica 9:	Descrição da característica 9
Característica 10:	Descrição da característica 10
Característica 11:	Descrição da característica 11
Característica 12:	Descrição da característica 12
Característica 13:	Descrição da característica 13
Característica 14:	Descrição da característica 14
Característica 15:	Descrição da característica 15

Apêndice D – Métricas Adotadas Nos Requisitos Não Funcionais

Funcionalidade: existência de um conjunto de funções que atendam às necessidades explícitas ou implícitas do usuário, quando a ferramenta estiver sendo utilizada sob as condições especificadas							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Adequação	Adequação Funcional	Quão adequada são as funções avaliadas?	X=1-A/B; onde A = Número de funções nas quais problemas são detectados na avaliação; e B = Número de funções avaliadas	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 – 1
Acurácia (Exatidão)	Expectativa para Acurácia	As diferenças entre os resultados esperados e os aceitáveis são aceitáveis?	X=A / T; onde A= Número de casos encontrados pelos usuários que atendem às expectativas da área de processo; e T= Número de requisitos da área de processo	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 – 1
Segurança de Acesso	Controlabilidade de Acesso	Quão controlável é o acesso ao sistema?	X= A / B; onde A= Número de tipos diferentes detectados de operações ilegais; e B= Número de tipos de operações ilegais da especificação	0 <= X <= 1; Quanto mais perto de 1, melhor	1	0,5	0

Interoperabilidade : capacidade da ferramenta de interagir com uma ou mais ferramentas especificadas							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Integração de Dados	Intercâmbio de Dados (baseado no formato dos dados)	Quão corretamente as interfaces de troca para determinados dados foram implementadas?	X= A / B; onde A= Número de formatos de dados gerados corretamente; e B= Número de formatos de dados que podem ser intercambiáveis	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 - 1,0
Integração de Interface	-	-	X= A / B; onde A= Quantidade de funcionalidades com interfaces semelhantes em outras ferramentas previamente selecionadas; e B= Quantidade de funcionalidades estudadas	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 - 1
Integração de Controle	-	-	X= A / B; A= Número de ferramentas cujo intercâmbio de dados ocorreu de acordo com os resultados esperados; B = Número de ferramentas possíveis para intercâmbio de dados, segundo a especificação	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 – 1
Integração de Conhecimento	-	-	X= A/B; onde A=Número de metadados acessíveis; B= Número de metadados existentes	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 – 1

Confiabilidade: capacidade da ferramenta de manter um nível de desempenho especificado, quando usada sob as condições especificadas							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Maturidade	Densidade de Falhas	Quantas falhas foram descobertas durante o período de teste?	X= A/B; A = Número de funcionalidades com falhas detectadas; e B = Número de funcionalidades avaliadas	0<=X<=1; Quanto mais perto de 0, melhor	0,6 - 1	0,3 - 0,5	0 - 0,1
Tolerância a Falhas	Impedimento de operação incorreta?	Quantas funções estão implementadas com capacidade para evitar operações incorretas?	X =A/B; onde A= Número de funcionalidades que apresentam capacidade para evitar falhas críticas e severas; e B= Número de funcionalidades avaliadas	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 – 1
Recuperabilidade	Restaurabilidade	Quão capaz é o produto de se restaurar depois de eventos anormais ou em requisições?	X=A/B; onde A= Número de eventos com restauração bem sucedida; e B=Número de eventos anormais	0<=X<=1; Quanto mais perto de 1, melhor, pois o produto é capaz de se restaurar em maior número de casos.	0 - 0,4	0,5 - 0,7	0,8 – 1
Utilização de Rotinas de Backup	-	-	X, onde X=0 significa não disponibilidade e utilização de backup's; e X=1 significa disponibilidade e utilização de backup	0 ou 1	0	-	1

Usabilidade: capacidade da ferramenta de ser compreendida, aprendida e operada de forma atraente ao usuário, quando usada sob as condições especificadas							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Inteligibilidade	Inteligibilidade de Funções	Quantas funções o usuário consegue descrever corretamente?	X=A/B; onde A = Número de funcionalidades (ou tipos de funcionalidades) entendidas; e B = Número total de funcionalidades (ou tipos de funcionalidades)	0<=X<=1; Quanto mais perto de 1, melhor	0 -0,4	0,5 - 0,7	0,8 – 1
Apreensibilidade	Facilidade de Aprendizado da Função	Quanto tempo leva para o usuário aprender determinada função?	T= Média de tempo que leva o usuário para aprender a função	T>0; quanto menor o T, melhor	DEPENDE DO TAMANHO E DA COMPLEXIDADE DA FERRAMENTA. DEVE SER ANALISADA CASO A CASO O NÍVEL DE ACEITAÇÃO PARA CADA FERRAMENTA		
Operacionalidade	Recuperabilidade de Erro Operacional em uso	O usuário pode recuperar-se de sua pior situação?	X = 1 - A/B; onde A= Número de situações de recuperação mal-sucedidas (depois de um erro ou alteração) nos quais os usuários não foram informados do risco pelo sistema; e B=Número de erros ou alterações do usuário	0<=X<= 1 Quanto mais perto de 1, melhor	0 - 0,4	0,5 - 0,7	0,8 – 1
Atratividade	Atratividade da Interação	Quão atrativa é a interface para os usuários?	X = A, onde A é uma nota de 5 a 10 para a facilidade de manipulação da interface	X>=5; Quanto mais perto de 10, melhor	5 - 6,9	7 - 8,5	8,6 – 10

Eficiência: capacidade da ferramenta de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob as condições especificadas							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Comportamento em relação ao Tempo	Tempo de Resposta	Quanto tempo leva para completar uma tarefa específica?	T = ARRED (Tr - Tf); onde Tr = Tempo de obter a resposta; e Tf = Tempo de entrada finalizada; e ARRED é o número arredondado (sem casas decimais)	T>=0; Quanto menor o tempo, melhor	>=6	2 - 5	0 – 1
Utilização de Recursos	Utilização de Memória Máxima	Qual o limite absoluto de memória requisitada para satisfazer determinada função?	X = Amax - Rmax; onde Amax = Memória máxima requisitada na avaliação; Rmax = Memória obtida antes do início da avaliação;	X>0; Quanto menor o X, melhor	DEPENDE DO TAMANHO E DA COMPLEXIDADE DA FERRAMENTA. DEVE SER ANALISADA CASO A CASO O NÍVEL DE ACEITAÇÃO PARA CADA FERRAMENTA		

Manutenibilidade: capacidade da ferramenta de ser modificada. As modificações podem incluir correções, melhorias ou adaptações, devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Analisabilidade	Capacidade de Análise de Falhas	O usuário pode identificar operações específicas que causaram falhas?	X= 1- A/B; onde A= Número de funcionalidades cujas falhas podem ser reproduzidas; e B= Número funcionalidades que causaram falhas no tempo de avaliação	0<=X<= 1 Quanto mais perto de 0, melhor	0,6-1	0,3-0,5	0-0,2
Modificabilidade	Capacidade de Controle de Alterações do Software	O mantenedor pode facilmente alterar um software para resolver problemas?	X= A/B; A= Número de alterações já registradas; e B= Número de atualizações. Atualizações se referem às distribuições das alterações. Caso a ferramenta não tenha política de atualização, ou não tenha havido atualização, ou a ferramenta não disponibilize os dados A e B, valor é 0.	X >= 0. Quanto mais perto de 0, então menor a quantidade de alterações ou maior a quantidade de atualizações. Quanto mais perto de A, menor a quantidade de atualizações.	DEVE SER VERIFICADO PARA CADA FERRAMENTA		
Estabilidade	Localização do Impacto da Modificação	O mantenedor pode mitigiar falhas causadas pela modificação?	X= A/N; onde A=Número de falhas que surgiram depois que uma falha foi resolvida; e N= Número de falhas resolvidas depois de alguma atualização. Caso não tenha havido atualização, o resultado é -1.	X>=0; Quanto mais perto de 0, melhor	>4 ou -1	1 - 3	0

Manutenibilidade: capacidade da ferramenta de ser modificada. As modificações podem incluir correções, melhorias ou adaptações, devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais (Cont.)							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Testabilidade	Eficiência do re-teste	O usuário e o mantenedor podem executar testes operacionais e determinar onde o software está ou não pronto para operação?	X= Sum(T)/N; onde T= Tempo gasto para testar e verificar se a falha foi ou não resolvida (unidade de tempo em horas); e N= Número de falhas resolvidas. Caso não tenha havido nenhuma atualização, o valor é -1.	X>=0; Quanto mais perto de 0, melhor	0,8 – 1	0,4 - 0,7	0,1 - 0,3
Frequência de Atualizações	-	-	X, onde X=0 significa que não existe política de atualização; e X=1 existe a política de atualização	0 ou 1	0	-	1

Portabilidade: capacidade da ferramenta de ser transferida de um ambiente para outro							
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**FAIXA DE ACEITAÇÃO**		
					Não Atende	**Atende Parcialmente**	**Atende Totalmente**
Adaptabilidade	Facilidade de Portabilidade para o usuário	O usuário ou mantenedor pode facilmente adaptar o software para o ambiente?	X= 1 - A/B; onde A= Número de operações nos quais as tarefas não foram completadas devido ao hardware disponibilizado; e B= Número total de funções que foram testadas	0<=X<=1; Quanto mais perto de 1, melhor	0-0,4	0,5-0,8	0,9-1
Capacidade para ser instalado	Facilidade de Instalação	O usuário pode facilmente instalar o software para o ambiente de operação?	X = 0, 1, 2, 3 ou 4; 0 - Não foi possível instalar a ferramenta no ambiente alvo; 1 - Houveram algumas dificuldades/problemas e a ferramenta não disponibiliza personalização de instalação; 2 - Houveram algumas dificuldades/problemas e a ferramenta disponibiliza personalização de instalação; 3 - Não houve quaisquer dificuldades/problemas e a ferramenta não disponibiliza personalização da instalação; 4 - Não houve quaisquer dificuldades/problemas e a ferramenta disponibiliza personalização da instalação	X=4 é o melhor resultado	0	1 - 3	4

Portabilidade: capacidade da ferramenta de ser transferida de um ambiente para outro (Cont.)							
					FAIXA DE ACEITAÇÃO		
Sub-Característica	**Métrica Utilizada**	**Definição Métrica**	**Fórmula - Utilizada**	**Faixa Aceitação**	**Não Atende**	**Atende Parcialmente**	**Atende Totalmente**
Coexistência	Disponibilidade de Co-Existência	Qual a média de tempo que o usuário leva para encontrar quaisquer restrições ou falhas inesperadas quando da operação concorrente com outros softwares?!	X = A/T; onde A = Número de restrições quaisquer ou falhas inesperadas, quando usado em concorrência a outro software; e T = Duração de operação com os software concorrentes	X>=0; Quanto mais perto de 0, melhor	Acima de 4	1 - 3	0 – 1
Capacidade para substituir	Inclusão de funções	O usuário ou mantenedor pode continuar a usar funções similares depois de substituir o software?	X = A/B; A = Número de funções que produzem resultados similares em ferramentas correlatas; e B = Número de funcionalidades da ferramenta	0<=X<=1; Quanto mais perto de 1, melhor	0 - 0,4	0,5 – 0,6	0,7 – 1

Apêndice E – Avaliação de Requisitos não Funcionais

LEGENDA:

1. *A, B, T*, ou outras variações, são as variáveis da fórmula da metrica
2. *TOTAL* é o resultado da aplicação da fórmula da métrica
3. *Res.* é o resultado final para a métrica, o que significa seu peso, obtido dos questionários, multiplicado pelo total encontrado na fórmula

Tabela E.1 - Avaliação de Requisitos Não Funcionais para Gerência de Requisitos

Funcionalidade: existência de um conjunto de funções que atendam às necessidades explícitas ou implícitas do usuário, quando a ferramenta estiver sendo utilizada sob as condições especificadas													
Sub-Característica	**PESO**	**RESULTADOS**											
		RTH				**Controla**				**OSRMT**			
Adequação	3,3	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		3	19	0,8	2	0	21	1	2	2	19	0,9	2
Acurácia (Exatidão)	3,8	**A**	**T**	**TOTAL**	**Res.**	**A**	**T**	**TOTAL**	**Res.**	**A**	**T**	**TOTAL**	**Res.**
		11	11	1	2	10	11	0,91	2	11	11	1	2
Segurança de Acesso	2,8	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		0	2	0	2	1	2	0,5	1	0	2	0	2
				TOTAL	19,8			**TOTAL**	17			**TOTAL**	19,8

Tabela E.1 - Avaliação de Requisitos Não Funcionais para Gerência de Requisitos (Cont.)

Interoperabilidade: capacidade da ferramenta de interagir com uma ou mais ferramentas especificadas													
Sub-Característica	**PESO**	**RESULTADOS**											
		RTH				**Controla**				**OSRMT**			
Integração de Dados	3,2	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		2	2	1	2	11	11	1	2	0	1	0	0
Integração de Interface	1,8	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		0	3	0	0	0	3	0	0	0	3	0	0
Integração de Controle	2,3	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		0	0	0	0	0	0	0	0	0	0	0	0
Integração de Conhecim.	2,3	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		7	7	1	2	7	7	1	2	5	5	1	2
				TOTAL	11			**TOTAL**	11			**TOTAL**	4,6

Confiabilidade: capacidade da ferramenta de manter um nível de desempenho especificado, quando usada sob as condições especificadas													
Sub-Característica	**PESO**	**RESULTADOS**											
		RTH				**Controla**				**OSRMT**			
Maturidade	3,2	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		3	19	0,16	1	3	21	0,14	1	2	19	0,11	1
Tolerância a Falhas	2,6	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		17	19	0,89	2	17	21	0,81	2	15	19	0,79	1
Recuperabilidade	3,1	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		0	2	0	0	0	1	0	0	0	1	0	0
Utilização de Rotinas de Backup	2,9	**X**			**Res.**	**X**			**Res.**	**X**			**Res.**
		0			0	0			0	0			0
				TOTAL	8,4			**TOTAL**	8,4			**TOTAL**	5,8

Tabela E.1 - Avaliação de Requisitos Não Funcionais para Gerência de Requisitos (Cont.)

Usabilidade: capacidade da ferramenta de ser compreendida, aprendida e operada de forma atraente ao usuário, quando usada sob as condições especificadas

Sub-Característica	PESO	RESULTADOS											
		RTH				Controla				OSRMT			
Inteligibilidade	2,1	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		7	7	1	2	7	7	1	2	6	6	1	2
Apreensibilidade	2,7	**T (min)**	**TOTAL MIN**	**TOTAL MAX**	**Res.**	**T (min)**	**TOTAL MIN**	**TOTAL MAX**	**Res.**	**T (min)**	**TOTAL MIN**	**TOTAL MAX**	**Res.**
		60	50	70	1	30	30	40	2	50	35	45	0
Operacionalidade	2,9	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		2	5	0,6	1	0	4	1	2	1	4	0,75	1
Atratividade	1,9	**A**		**TOTAL**	**Res.**	**A**		**TOTAL**	**Res.**	**A**		**TOTAL**	**Res.**
		7		7	1	9		9	2	7		7	1
		TOTAL			11,7	**TOTAL**			19,2	**TOTAL**			9

Eficiência: capacidade da ferramenta de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob as condições especificadas

Sub-Característica	PESO	RESULTADOS											
		RTH				Controla				OSRMT			
Comportamento em relação ao Tempo	3,1	**T**		**TOTAL**	**Res.**	**T**		**TOTAL**	**Res.**	**T**		**TOTAL**	**Res.**
		0		0	2	0		0	2	0		0	2
Utilização de Recursos	2,7	**Amax**	**Rmax**	**TOTAL**	**Res.**	**Amax**	**Rmax**	**TOTAL**	**Res.**	**Amax**	**Rmax**	**TOTAL**	**Res.**
		257	246	11	2	250	230	20	2	275	230	45	0
		TOTAL			11,6	**TOTAL**			11,6	**TOTAL**			6,2

Tabela E.1 - Avaliação de Requisitos Não Funcionais para Gerência de Requisitos (Cont.)

Manutenibilidade: capacidade da ferramenta de ser modificada. As modificações podem incluir correções, melhorias ou adaptações, devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais													
Sub-Característica	**PESO**	**RESULTADOS**											
		RTH				**Controla**				**OSRMT**			
Analisabilidade	2,7	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		3	3	0	2	0	0	0	2	2	2	0	2
Modificabilidade	3,1	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		0	0	0	1	0	0	0	1	112	5	22,4	2
Estabilidade	3,1	**A**	**N**	**TOTAL**	**Res.**	**A**	**N**	**TOTAL**	**Res.**	**A**	**N**	**TOTAL**	**Res.**
		0	0	0	2	0	0	0	2	0	0	0	2
Testabilidade	2,9	**Sum(T)**	**N**	**TOTAL**	**Res.**	**Sum(T)**	**N**	**TOTAL**	**Res.**	**Sum(T)**	**N**	**TOTAL**	**Res.**
		0	0	0	2	0	0	0	2	0	0	0	2
Frequência de Atualizações	2,7	**X**		**TOTAL**	**Res.**	**T**		**TOTAL**	**Res.**	**T**		**TOTAL**	**Res.**
		1		1	2	1		1	2	1		1	2
		TOTAL			25,9	**TOTAL**			25,9	**TOTAL**			29

Tabela E.1 - Avaliação de Requisitos Não Funcionais para Gerência de Requisitos (Cont.)

Portabilidade: capacidade da ferramenta de ser transferida de um ambiente para outro													
Sub-Característica	**PESO**	**RESULTADOS**											
		RTH				**Controla**				**OSRMT**			
Adaptabilidade	2,4	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		0	19	1	2	0	21	1	2	0	19	1	2
Capacidade para ser instalado	3,2	**X**			**Res.**	**X**			**Res.**	**X**			**Res.**
		2			1	3			1	2			1
Coexistência	3,4	**A**	**T**	**TOTAL**	**Res.**	**A**	**T**	**TOTAL**	**Res.**	**A**	**T**	**TOTAL**	**Res.**
		0	30	0	2	0	30	0	2	0	30	0	2
Capacidade para substituir	2,4	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**	**A**	**B**	**TOTAL**	**Res.**
		5	7	0,71	2	6	11	0,55	1	5	7	0,71	2
		TOTAL			19,6	**TOTAL**			17,2	**TOTAL**			19,6
	TOTAL GERAL	**108**				**110,3**				**94**			

Tabela E.2 - Avaliação de Requisitos Não Funcionais para Gerência de Projetos

Funcionalidade: existência de um conjunto de funções que atendam às necessidades explícitas ou implícitas do usuário, quando a ferramenta estiver sendo utilizada sob as condições especificadas

Sub-Carac.	PESO	RESULTADOS																							
		DotProject				NetOffice				GPTool				Open WorkBench				PHProjekt				Streber			
Ade-quação	3,3	A	B	TOT	Res	A	B	TOT	Res	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		4	21	0,8	2	1	18	0,9	2	1	8	0,9	2	1	11	0,91	2	4	15	0,73	1	4	12	0,67	1
Acurácia (Exatidão)	3,8	A	T	TOT	Res.	A	T	TOT	Res.	A	T	TOT	Res.	A	T	TOT.	Res.	A	T	TOT.	Res.	A	T	TOT.	Res.
		37	39	0,95	2	38	39	0,97	2	24	39	0,62	1	32	39	0,82	2	36	39	0,92	2	35	39	0,9	2
Segu-rança de Acesso	2,8	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		0	2	0	2	0	2	0	2	2	2	1	0	2	2	1	0	1	2	0,5	1	0	2	0	2
		TOTAL			19,8	TOTAL			19,8	TOTAL			10,4	TOTAL			14,2	TOTAL			13,7	TOTAL			16,5

Interoperabilidade: capacidade da ferramenta de interagir com uma ou mais ferramentas especificadas

Sub-Carac.	PESO	RESULTADOS																							
		DotProject				NetOffice				GPTool				Open WorkBench				PHProjekt				Streber			
Integ. Dados	3,2	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		5	5	1	2	8	8	1	2	5	7	0,71	1	0	0	0	0	5	8	0,63	1	0	0	0	0
Integ. Inter Face	1,8	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		3	3	1	2	3	3	1	2	0	3	0	0	2	3	0,67	1	2	3	0,67	1	3	3	1	2
Integ. Controle	2,3	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Integ. Conhec	2,3	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		8	8	1	2	6	6	1	2	2	3	0,67	1	4	4	1	2	6	8	0,75	1	5	5	1	2
		TOTAL			19,2	TOTAL			14,6	TOTAL			5,5	TOTAL			6,4	TOTAL			7,3	TOTAL			8,2

Tabela E.2 - Avaliação de Requisitos Não Funcionais para Gerência de Projetos (Cont.)

Confiabilidade: capacidade da ferramenta de manter um nível de desempenho especificado, quando usada sob as condições especificadas

Sub-Carac.	PESO	RESULTADOS																							
		DotProject				NetOffice				GPTool				Open WorkBench				PHProjekt				Streber			
Maturidade	3,2	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		4	21	0,19	1	1	18	0,06	2	1	8	0,13	1	1	11	0,09	2	4	15	0,27	1	0	12	0	2
Tolerância Falhas	2,6	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		19	21	0,9	2	17	18	0,94	2	7	8	0,88	2	11	11	1	2	14	15	0,93	2	12	12	1	2
Recuperabilidade	3,1	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		0	3	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Rotina Bkp	2,9	X			Res.	X			Res.	X			Res.	X			Res.	X			Res.	X			Res.
		1			2	1			2	0			0	0			0	0			0	0			0
		TOTAL			14,2	TOTAL			17,4	TOTAL			8,4	TOTAL			11,6	TOTAL			8,4	TOTAL			11,6

Usabilidade: capacidade da ferramenta de ser compreendida, aprendida e operada de forma atraente ao usuário, quando usada sob as condições especificadas

Sub-Carac.	PESO	RESULTADOS																							
		DotProject				NetOffice				GPTool				Open WorkBench				PHProjekt				Streber			
Inteligibilidade	2,1	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		21	21	1	2	18	18	1	2	7	8	0,88	2	7	8	0,875	2	11	15	0,73	1	12	12	1	2
Apreensibilidade	2,7	T (min)	TOTAL MIN	TOT MAX	Res.	T (min)	TOTAL MIN	TOT MAX	Res.	T (min)	TOTAL MIN	TOT MAX	Res.	T (min)	TOTAL MIN	TOT MAX	Res.	T (min)	TOTAL MIN	TOT MAX	Res.	T (min)	TOTAL MIN	TOT MAX	Res.
		50	60	80	2	50	60	80	2	30	25	35	1	30	30	40	2	40	35	45	1	30	35	45	2
Operacionalidade	2,9	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		1	4	0,75	1	0	4	1	2	3	3	0	0	2	3	0,33	0	2	4	0,5	1	4	4	0	0
Atratividade	1,9	X		TOT	Res.	X		TOT	Res.	X		TOT	Res.	X		TOT	Res.	X		TOT	Res.	X		TOT	Res.
		8		8	1	9		9	2	7		7	1	8		8	1	6		6	0	9		9	2
		TOTAL			14,4	TOTAL			19,2	TOTAL			8,8	TOTAL			11,5	TOTAL			7,7	TOTAL			13,4

Tabela E.2 - Avaliação de Requisitos Não Funcionais para Gerência de Projetos (Cont.)

Eficiência: capacidade da ferramenta de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob as condições especificadas

Sub-Carac.	PESO	RESULTADOS																							
		DotProject				NetOffice				GPTool				Open WorkBench				PHProjekt				Streber			
Comp. Relação Tempo	3,1	T		TOT	Res.	T		TOT	Res.	T		TOT	Res.	T		TOT	Res.	T		TOT	Res.	T		TOT	Res.
		0		0	2	0		0	2	0		0	2	0		0	2	0		0	2	0		0	2
Uso de Recurso	2,7	Amax	Rmax	TOT	Res.	Amax	Rmax	TOT	Res.	Amax	Rmax	TOT	Res.	Amax	Rmax	TOT	Res.	Amax	Rmax	TOT	Res.	Amax	Rmax	TOT	Res.
		270	258	12	2	265	260	5	2	263	233	30	1	288	233	55	0	290	263	27	1	270	262	8	2
		TOTAL			11,6	TOTAL			11,6	TOTAL			8,9	TOTAL			6,2	TOTAL			8,9	TOTAL			11,6

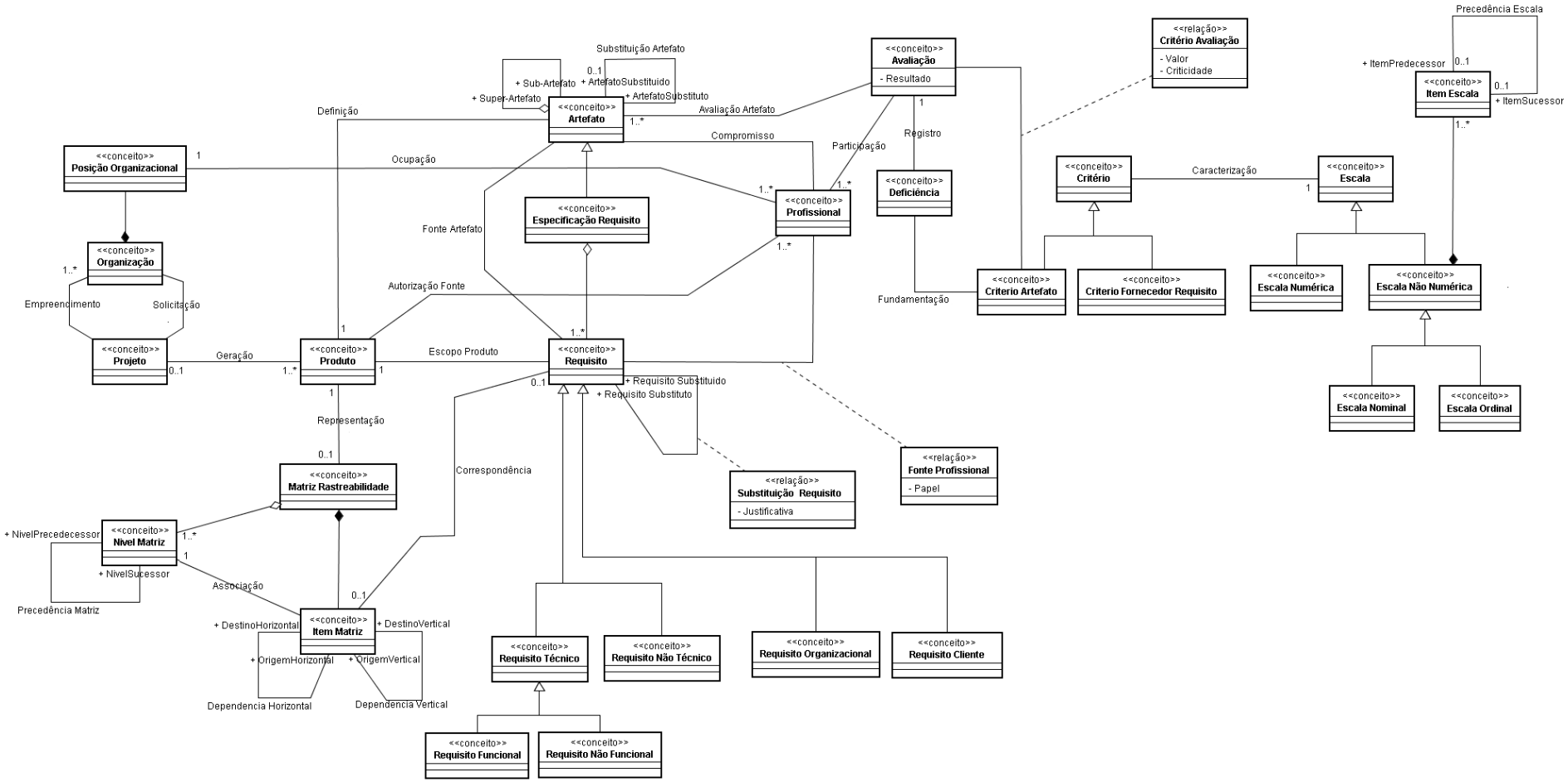
Manutenibilidade: capacidade da ferramenta de ser modificada. As modificações podem incluir correções, melhorias ou adaptações, devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais

Sub-Carac.	PESO	RESULTADOS																							
		DotProject				NetOffice				GPTool				Open WorkBench				PHProjekt				Streber			
Analisablidade	2,7	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		4	4	0	2	1	1	0	2	1	1	0	2	1	1	0	2	4	4	0	2	0	0	0	2
Modificabilidade	3,1	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.	A	B	TOT	Res.
		593	12	49,4	2	17	5	3,4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Estabilidade	3,1	A	N	TOT	Res.	A	N	TOT	Res.	A	N	TOT	Res.	A	N	TOT	Res.	A	N	TOT	Res.	A	N	TOT	Res.
		0	1	0	2	0	1	0	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	0	2
Testabilidade	2,9	Sum(T)	N	TOT	Res.	Sum(T)	N	TOT	Res.	Sum(T)	N	TOT	Res.	Sum(T)	N	TOT	Res.	Sum(T)	N	TOT	Res.	Sum(T)	N	TOT	Res.
		0	0	0	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	0	2	0	0	0	2
Freq Atualiz	2,7	X		TOT	Res.	T		TOT	Res.	T		TOT	Res.	T		TOT	Res.	T		TOT	Res.	T		TOT	Res.
		1		1	2	1		1	2	1		1	2	1		1	2	1		1	2	1		1	2
		TOTAL			29	TOTAL			25,9	TOTAL			22,8	TOTAL			22,8	TOTAL			22,8	TOTAL			22,8

Tabela E.2 - Avaliação de Requisitos Não Funcionais para Gerência de Projetos (Cont.)

Portabilidade: capacidade da ferramenta de ser transferida de um ambiente para outro																									
Sub-Carac.	**PESO**	**RESULTADOS**																							
		DotProject				**NetOffice**				**GPTool**				**Open WorkBench**				**PHProjekt**				**Streber**			
Adaptabilidade	2,4	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**
		0	21	1	2	0	18	1	2	0	8	1	2	0	11	1	2	0	15	1	2	0	12	1	2
Capac. Instal.	3,2		**X**		**Res.**		**X**		**Res.**		**X**		**Res.**		**X**		**Res.**		**X**		**Res.**		**X**		**Res.**
			4		2		4		2		4		2		4		2		2		1		4		2
Coexistência	3,4	**A**	**T**	**TOT**	**Res.**	**A**	**T**	**TOT**	**Res.**	**A**	**T**	**TOT**	**Res.**	**A**	**T**	**TOT**	**Res.**	**A**	**T**	**TOT**	**Res.**	**A**	**T**	**TOT**	**Res.**
		0	30	0	2	0	30	0	2	0	30	0	2	0	30	0	2	0	30	0	2	0	30	0	2
Capacidade Subst.	2,4	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**	**A**	**B**	**TOT**	**Res.**
		4	4	1	2	4	4	1	2	2	4	0,5	1	2	4	2	2	4	4	1	2	4	4	1	2
		TOTAL			22,8	**TOTAL**			22,8	**TOTAL**			20,4	**TOTAL**			22,8	**TOTAL**			19,6	**TOTAL**			22,8
TOTAL GERAL		**131**				**131,3**				**85,2**				**95,5**				**88,4**				**106,9**			

Apêndice F – Modelo Completo da Ontologia para Gerência de Requisitos



Anexo A – Linguagem para Modelagem de Processo

A linguagem descrita a seguir foi proposta por (VILLELA, 2004) para modelagem de processos. Ela é composta de elementos gráficos que podem ser do tipo área, objeto ou ligação, onde uma ligação estabelece uma relação entre dois objetos e uma área agrupa objetos, definindo um contexto para os mesmos. Objetos ainda permitem adornos, utilizados para representar explicitamente características dos objetos. A seguir, são apresentados apenas elementos da linguagem relevantes para este trabalho.

Objeto	Notação	Definição
Atividade		Objeto que informa a presença de uma única atividade no fluxo do processo.
Estado Inicial		Iníco do fluxo de atividades
Estado Inicial		Fim do fluxo de atividades

Quadro A.1 – Definição e Notação dos Objetos

Objeto	Notação com Adornos	Definição dos Adornos
Atividade	(a) (b)	(a) Adorno que indica que a atividade é composta, ou seja, ela pode ser decomposta em sub-atividades; (b) Adorno que indica que a atividade é opcional, ou seja, sua execução não é obrigatória
Operação Lógica	(a) (b) (c) (d)	(a) Adorno que indica a operação lógica E; (b) Adorno que indica a operação lógica OU; (c) Adorno que indica a operação lógica OU Exclusivo; (c) Notação sem adorno, que é utilizada como elemento de junção após uso de uma operação lógica que atua como elemento de divisão.

Quadro A.2 – Definição e Notação dos Adornos