
Um processo envolvendo elementos de
design rationale **para projetos de pesquisa em**
software

Débora Maria Barroso Paiva

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data da Defesa : 14.06.2007

Visto do orientador: _____

Um processo envolvendo elementos de *design rationale* para projetos de pesquisa em software

Débora Maria Barroso Paiva

Orientadora: *Profa. Dra. Renata Pontin de Mattos Fortes*

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional.

“VERSÃO REVISADA APÓS A DEFESA”

ICMC - USP - São Carlos
Maio de 2007

À minha querida mãe, pela preciosidade de seu coração!

Agradecimentos

A Deus, por iluminar o meu caminho.

À minha mãe, por seu amor incondicional, por ter me ensinado a batalhar pelos sonhos e a ser forte perante as dificuldades.

À minha irmã pela amizade, pelo imenso amor que tem por mim, pelo companheirismo e por me apoiar sempre.

Aos meus irmãos, pelo carinho.

Aos meus sobrinhos, pelos momentos de alegria tão especiais e renovadores.

À minha orientadora, professora Dra. Renata Pontin de Mattos Fortes, pelos ensinamentos, pelas partilhas e, sobretudo, pela amizade e pela confiança que depositou em mim como sua primeira aluna de doutorado.

À professora Dra. Rosely Sanches pelas inúmeras e valiosas contribuições durante o desenvolvimento do trabalho, pelo carinho e por transmitir uma energia tão positiva.

Ao professor Dr. José Carlos Maldonado, pelas excelentes sugestões e críticas em relação ao trabalho, pela disponibilidade em colaborar e pela amizade.

Aos meus familiares que sempre me incentivaram e apoiaram.

Aos vários amigos e colegas, pessoas muito especiais, com os quais convivi nesse período. Aprendi muito com vocês! Obrigada por terem ajudado a renovar a esperança nos momentos difíceis e por terem compartilhado os momentos alegres.

A todos do LABES e do grupo de pesquisa, pela amizade, pelo agradável convívio, pelas discussões e trocas de experiências tão enriquecedoras. Em especial ao André Pimenta Freire, pelas expressivas contribuições no trabalho.

Aos professores, funcionários e estudantes do Departamento de Computação e Estatística da Universidade Federal de Mato Grosso do Sul, por terem me recebido com tanto carinho.

A todos aqueles que participaram dos estudos de caso realizados, pelas importantes contribuições.

Ao ICMC-USP pela oportunidade e à CAPES pelo auxílio financeiro.

Resumo

O volume crescente de projetos de pesquisa que envolvem software motiva a realização de análises sobre os processos utilizados no desenvolvimento desses projetos, considerando-se as atividades realizadas, os resultados obtidos e a aplicação de recursos. Na área de Engenharia de Software, processos têm sido propostos nos últimos anos com o objetivo de contribuir para o desenvolvimento de projetos de pesquisa que envolvem software. Busca-se, de forma geral, auxiliar o gerenciamento dos diversos artefatos que podem ser gerados, por exemplo, modelos, código fonte, relatórios técnicos e artigos científicos. As soluções apresentadas na literatura começaram a ser propostas recentemente e, portanto, os estudos realizados nessa área apresentam-se em estágio bastante inicial. Dessa forma, o objetivo deste trabalho foi definir um processo para o desenvolvimento de projetos de pesquisa envolvendo software, que esteja de acordo com um conjunto de requisitos identificado e que possa contribuir para a evolução desses projetos. Para reforçar a continuidade dos projetos, foi estudada a abordagem de *design rationale*, com o objetivo de oferecer meios para que as decisões sejam capturadas e registradas em fases específicas do desenvolvimento. O processo de documentação foi enfatizado, ou seja, foi investigado como a abordagem de *design rationale* poderia ser utilizada para melhorar a documentação dos projetos de pesquisa. Como um resultado obtido, foi definido um modelo para representação de *design rationale* (DR-SACI), implementado em uma ferramenta CASE e avaliado em um experimento.

Abstract

The increasing volume of research projects in the context of software is a motivation for analyzing processes which are used to develop such projects, regarding activities that are carried out, results that are gathered and application of resources. In the Software Engineering area, recently, processes have been considered with the objective to contribute for research projects development in which software is an element. Overall, the main purpose is to help management overtime of artifacts that can be produced, such as, models, code, technical reports and scientific papers. However, research developed in this direction and solutions discussed in the literature are presented in an initial stage. Therefore, the objective of this work was to define a process for development of research projects regarding a set of requirements. It is expected that such process can be useful to contribute for evolution of these projects. To reinforce the aspect of research project evolution, the design rationale approach was studied. The goal was to provide the opportunity of capturing and registering decisions in specific development phases. The documentation process was emphasized, i.e., the design rationale approach was analyzed focusing on the improvement of research projects documentation. As a result, a model for design rationale representation was defined, implemented in a CASE tool and evaluated by means of an experiment.

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Considerações iniciais	1
1.2 Contexto e motivação	3
1.3 Objetivos e principal contribuição	5
1.4 Organização da tese	6
2 Fundamentos sobre Processos	7
2.1 Considerações iniciais	7
2.2 Visão geral sobre processos de software	7
2.3 Desenvolvimento distribuído de projetos	10
2.4 Desenvolvimento colaborativo de projetos	11
2.5 Processo de software livre	13
2.6 Métodos ágeis	15
2.7 Qualidade de processos de software	18
2.8 Processos e ambientes para desenvolvimento de projetos de pesquisa	21
2.8.1 <i>eXtreme Researching</i>	22
2.8.2 Processo padrão para pesquisa e desenvolvimento	24
2.8.3 HDG (<i>Higher Degree Process</i>)	26
2.8.4 A proposta de Nunamaker e Chen	27
2.8.5 <i>Collaboratories</i>	28
2.9 Considerações finais	30
3 <i>Design Rationale</i>	31
3.1 Considerações iniciais	31
3.2 Visão geral sobre <i>design rationale</i>	31
3.3 Atividades envolvendo <i>design rationale</i>	33
3.3.1 Captura de <i>design rationale</i>	33
3.3.2 Representação de <i>design rationale</i>	35
3.3.3 Recuperação de <i>design rationale</i>	39
3.4 Uso de <i>design rationale</i>	41
3.4.1 Visando colaboração	41
3.4.2 Visando manutenção e reuso	42
3.4.3 Visando a melhoria de qualidade	43
3.4.4 Visando a transferência de conhecimento	44
3.5 <i>Design rationale</i> no desenvolvimento de projetos de pesquisa	44
3.6 Limitações, perspectivas e desafios em <i>design rationale</i>	46

3.7	Considerações finais	47
4	Concepção de um Processo para Projetos de Pesquisa em Software	49
4.1	Considerações iniciais	49
4.2	Proposta de Humphrey para definição de processos de software	49
4.3	Determinação das características do processo	51
4.4	Definição de objetivos, metas e critérios de qualidade do processo	52
4.5	Caracterização do processo corrente	58
4.6	Caracterização do processo almejado	61
4.7	Estabelecimento de uma estratégia de desenvolvimento do processo	62
4.7.1	Abordagem para apresentação do processo	63
4.7.2	Estudos de caso: documentação, manutenção e comunicação	63
4.7.3	Estudo de caso: avaliação e melhoria do processo utilizado no projeto <i>No-Risk Planning</i>	67
4.8	Considerações finais	69
5	Processo para Projetos de Pesquisa em Software	71
5.1	Considerações iniciais	71
5.2	Fundamentos para a elaboração do processo proposto	72
5.3	Processo padrão para desenvolvimento de projetos de pesquisa em software	73
5.4	Especialização do processo padrão para projetos de pesquisa em software .	75
5.5	Instanciação do processo padrão para projetos de pesquisa em software . .	75
5.6	Avaliação do processo padrão para projetos de pesquisa em software	81
5.6.1	Em relação a requisitos, características e propriedades de processos	81
5.6.2	Em relação ao estado da arte	82
5.7	Considerações finais	83
6	Um Modelo de <i>Design Rationale</i> para Projetos de Pesquisa em Software	85
6.1	Considerações iniciais	85
6.2	Experiências com <i>design rationale</i> em ambiente acadêmico	85
6.3	Modelo de representação de <i>design rationale</i> para desenvolvimento de projetos de pesquisa em software	87
6.4	Ferramenta para apoiar o registro de <i>design rationale</i> de projetos de pesquisa em software	91
6.5	Experimento envolvendo o modelo de <i>design rationale</i> e a ferramenta MV-CASE	93
6.5.1	Definição do experimento	93
6.5.2	Planejamento do experimento	94
6.5.3	Realização do experimento	98
6.5.4	Resultados dos experimentos	99
6.6	Comparação entre propostas para registro de <i>design rationale</i> em projetos de pesquisa	102
6.7	Considerações finais	103
7	Conclusões	105
7.1	Contribuições	105
7.2	Limitações	106
7.3	Trabalhos futuros	107
7.4	Produção científica	108

Referências Bibliográficas	111
A Práticas de Desenvolvimento de Software em Projetos de Pesquisa	135
B Processo Padrão de Desenvolvimento de Projetos de Pesquisa	141
B.1 Processos fundamentais	141
B.1.1 Processo de Aquisição	141
B.1.2 Processo de Iniciação	144
B.1.3 Processo de Desenvolvimento	154
B.1.4 Processo de Operação	163
B.1.5 Processo de Manutenção	165
B.2 Processos de apoio	167
B.2.1 Processo de Documentação	167
B.2.2 Processo de Gerenciamento de Configuração	169
B.2.3 Processo de Garantia da Qualidade	172
B.2.4 Processo de Verificação	174
B.2.5 Processo de Validação	175
B.2.6 Processo de Revisão	176
B.2.7 Processo de Resolução de Problemas	179
B.2.8 Processo de Revisão Sistemática	180
B.2.9 Processo de Preparação de Documentos Científicos	183
B.2.10 Processo de Elaboração de Módulos Educacionais	185
B.2.11 Processo de <i>Postmortem</i>	187
B.2.12 Processo de Transferência Tecnológica	190
B.3 Processos organizacionais	194
B.3.1 Processo de Treinamento	195
B.3.2 Processo de Gerenciamento	197
B.3.3 Processo de Infra-estrutura	207
B.3.4 Processo de Melhoria	209
B.3.5 Processo de Planejamento	212
B.3.6 Processo de Divulgação	219
B.3.7 Processo de Comunicação	221
B.3.8 Processo de Coordenação	222
B.3.9 Processos para Estabelecimento de Parceria Universidade-Empresa	224

Lista de Figuras

2.1	Estrutura da Norma ISO/IEC 12207 (1995)	19
2.2	Visão geral do processo para pesquisa e desenvolvimento proposto por Hwang e Park (2006)	25
2.3	Processo para desenvolvimento de projetos de pesquisa proposto por Numanaker e Chen (1990)	27
3.1	Abstrações e relacionamentos do modelo IBIS (Kunz e Rittel, 1970) apud (Regli et al., 2000)	37
3.2	Abstrações e relacionamentos do modelo PHI (McCall, 1991)	37
3.3	Abstrações e relacionamentos do modelo QOC (MacLean et al., 1991)	38
3.4	Representação de <i>design rationale</i> de acordo com a proposta de Kim et al. (1993)	45
4.1	Práticas executadas no desenvolvimento de 36 projetos de pesquisa (em termos de porcentagem)	59
4.2	Definição de processo de software para equipes geograficamente distribuídas (Rocha et al., 2001a)	64
5.1	Processo padrão para o desenvolvimento de projetos de pesquisa em software	74
6.1	Modelo de representação de <i>design rationale</i> para o contexto de desenvolvimento de projetos de pesquisa em software (DR-SACI)	90
6.2	Exemplo de uso da ferramenta MVCASE em que informações de <i>design rationale</i> foram capturadas	92
6.3	Quantidade de itens de <i>design rationale</i> registrados em relação ao uso dos modelos IBIS e DR-SACI	99
6.4	Resultados do teste de normalidade dos dados obtidos com o uso do modelo IBIS	100
6.5	Resultados do teste de normalidade dos dados obtidos com o uso do modelo DR-SACI	101
6.6	Resultados da avaliação da estrutura do modelo DR-SACI	101
6.7	Resultados da avaliação da importância do modelo DR-SACI na fase de manutenção	102
6.8	Opinião dos estudantes que realizaram a manutenção sem o uso de <i>design rationale</i>	102

Lista de Tabelas

2.1	Práticas do método XR (Chirouze et al., 2005)	23
2.2	Práticas base do HDG (Walker, 2003)	26
5.1	Especialização do processo padrão para o nível 2 do CMMI. No Apêndice B encontram-se as descrições das atividades e tarefas que estão apresentadas nesta tabela por meio de seus identificadores.	76
5.2	Especialização do processo padrão para o nível 3 do CMMI. No Apêndice B encontram-se as descrições das atividades e tarefas que estão apresentadas nesta tabela por meio de seus identificadores.	77
6.1	Informações de <i>design rationale</i> registradas em 31 projetos acadêmicos de software (valores médios)	86
6.2	Projeto do experimento	95
6.3	Resultados quantitativos em relação à captura de <i>design rationale</i>	99
B.1	Informações adicionais relacionadas à tarefa ET.1	146
B.2	Informações adicionais relacionadas à tarefa ET.2	148
B.3	Informações adicionais relacionadas à tarefa ET.3	149
B.4	Informações adicionais relacionadas à tarefa AE.2	150
B.5	Informações adicionais relacionadas à tarefa EP.1	153
B.6	Informações adicionais relacionadas à tarefa DF.1	156
B.7	Informações adicionais relacionadas à tarefa PR.1	157
B.8	Informações adicionais relacionadas à tarefa PP.1	158
B.9	Informações adicionais relacionadas à tarefa DP.2	159
B.10	Informações adicionais relacionadas à tarefa DP.3	160
B.11	Informações adicionais relacionadas à tarefa AP.1	160
B.12	Informações adicionais relacionadas à tarefa IR.1	162
B.13	Informações adicionais relacionadas à tarefa RM.2	166
B.14	Informações adicionais relacionadas à tarefa RR.3	179
B.15	Informações adicionais relacionadas à tarefa RD.1	184
B.16	Informações adicionais relacionadas à tarefa EM.1	186
B.17	Informações adicionais relacionadas à tarefa EM.2	187
B.18	Informações adicionais relacionadas à tarefa EM.3	187
B.19	Informações adicionais relacionadas à tarefa PE.1	189
B.20	Informações adicionais relacionadas à tarefa FA.1	190
B.21	Informações adicionais relacionadas à tarefa PT.1	191
B.22	Informações adicionais relacionadas à tarefa RS.2	194
B.23	Informações adicionais relacionadas à tarefa TR.1	196
B.24	Informações adicionais relacionadas à tarefa GR.1	199
B.25	Informações adicionais relacionadas à tarefa GR.2	200

B.26	Informações adicionais relacionadas à tarefa GC.1	202
B.27	Alocação de papéis e responsabilidades aos membros	203
B.28	Informações adicionais relacionadas à tarefa GT.1	204
B.29	Informações adicionais relacionadas à tarefa GO.1	205
B.30	Informações adicionais relacionadas à tarefa GS.1	205
B.31	Informações adicionais relacionadas à tarefa DE.2	214
B.32	Informações adicionais relacionadas à tarefa PI.2	215
B.33	Informações adicionais relacionadas à tarefa PZ.1	216
B.34	Artefato sugerido para a atividade GM.2	218
B.35	Informações adicionais relacionadas à tarefa IA.1	220
B.36	Informações adicionais relacionadas à tarefa PC.2	223
B.37	Informações adicionais relacionadas à tarefa EA.1	225
B.38	Informações adicionais relacionadas à tarefa EA.2	226
B.39	Informações adicionais relacionadas à tarefa EA.3	227
B.40	Resumo dos identificadores definidos para os processos fundamentais do processo padrão de projetos de pesquisa em software	227
B.41	Resumo dos identificadores definidos para os processos de apoio do processo padrão de projetos de pesquisa em software	228
B.42	Resumo dos identificadores definidos para os processos organizacionais do processo padrão de projetos de pesquisa em software	230

Lista de Abreviaturas

CMMI: *Capability Maturity Model Integration*

CNPq: *Conselho Nacional de Desenvolvimento Científico e Tecnológico*

DR-SACI: *Design Rationale of Software Developed in Research Environment*

EAP: *Estrutura Analítica de Projeto*

Finep: *Financiadora de Estudos e Projetos*

GENESIS: *GENERALISED eNvironment for procEsS management in cooperatIve Software engineering*

IBIS: *Issue-Based Information System*

IEC: *International Electrotechnical Commission*

ISO: *International Organization for Standardization*

OWL: *Web Ontology Language*

P&D: *Pesquisa e Desenvolvimento*

PHI: *Procedural Hierarchy of Issues*

PML: *Process Modeling Language*

PSP: *Personal Software Process*

QOC: *Questions, Options and Criteria*

RDF: *Resource Description Framework*

SAFE: *Software Engineering Available for Everyone*

TSP: *Team Software Process*

XP: *Extreme Programming*

XR: *Extreme Researching*

Introdução

1.1 Considerações iniciais

Diversos modelos de processo têm sido propostos nos últimos anos, em diferentes áreas, com o objetivo de auxiliar as organizações a sistematizar e melhorar o cumprimento das atividades que levam à produção dos bens de interesse. Diferentes objetivos podem ser almeçados quando alguém se propõe a melhorar a qualidade dos processos, dentre eles, melhoria na produtividade e aumento da satisfação das pessoas envolvidas.

É notável que, nos últimos anos, os resultados obtidos com o desenvolvimento de projetos de pesquisa têm contribuído de forma significativa para o desenvolvimento científico e tecnológico em diversas áreas de conhecimento. À medida que os projetos se tornam mais complexos e passam a ser desenvolvidos de forma distribuída e colaborativa, nota-se que a definição, a implantação e a melhoria dos diversos processos cumpridos torna-se fundamental (Chirouze et al., 2005; Segal, 2005). Busca-se, com isso, estabelecer um ambiente favorável para que o conhecimento, o principal resultado que é almejado quando se desenvolve pesquisa, seja de fato adquirido.

Outro fator importante é que a qualidade dos artefatos produzidos durante o desenvolvimento de projetos de pesquisa (por exemplo, software, documentos técnicos, diagramas, modelos, relatórios e artigos científicos) é exigência fundamental de agências que financiam e de instituições que promovem o desenvolvimento de pesquisas, além de ser, no geral, uma exigência profissional dos próprios pesquisadores envolvidos. De acordo com Ambati e Kishore (2004), a definição, o uso e a melhoria contínua de um processo para o desenvolvimento de projetos de pesquisa em software são consideradas questões primordiais que precisam ser tratadas para que seja possível melhorar a qualidade dos resultados de pesquisas obtidos e a produtividade dos membros envolvidos.

Uma importante característica do contexto de desenvolvimento de pesquisa é a natureza colaborativa e distribuída dos projetos, que se tornou evidente principalmente com o advento da internet. Diversos projetos estão sendo desenvolvidos sob essas perspectivas (Aversano et al., 2004; Jenkins, 2004). Para exemplificar, o projeto COSPA (*Consortium for Open Source Software in the Public Administration*)(Succi e Zuliani, 2004) está sendo desenvolvido por quinze instituições da Europa, com participação de membros da academia, indústria e governo de seis diferentes países e tem como principal objetivo estudar como software livre pode ser utilizado na administração pública. A partir dos relatos dos participantes observa-se, de forma clara, a importância em considerar atividades que ajudem a promover o controle e o gerenciamento dos projetos e, sobretudo, a comunicação entre os membros das equipes que participam do desenvolvimento.

Outra característica importante que motiva a definição, o uso e a melhoria de processos para o desenvolvimento de pesquisas é o estabelecimento de parcerias entre indústria e academia. Em diversos países da Europa e nos Estados Unidos as contribuições provenientes dessas parcerias têm sido expressivas (Bleek et al., 2005; Kornecki et al., 2003). No Brasil o governo também está incentivando e participando de associações para o desenvolvimento de novas tecnologias. Os fundos setoriais de ciência e tecnologia da Finep¹ (Financiadora de Estudos e Projetos), por exemplo, são instrumentos de financiamento de projetos de pesquisa, desenvolvimento e inovação no país, sendo que um deles, denominado CT- Verde-Amarelo, se refere a um programa de estímulo à interação universidade-empresa para apoio à inovação tecnológica. O CNPq² (Conselho Nacional de Desenvolvimento Científico e Tecnológico) apresenta uma iniciativa semelhante ao promover a cooperação científica e tecnológica internacional e apoiar o desenvolvimento de projetos de pesquisas por membros de diferentes países³. A exigência pela qualidade nos resultados desses projetos é evidenciada nos editais de chamada à submissão de propostas e, geralmente, é fundamental que seja demonstrado o potencial para exportação dos resultados que se pretende obter. A formação de parcerias entre membros da indústria, academia e governo resulta em requisitos adicionais de um processo para desenvolvimento de pesquisas. Em particular, o apoio à transferência de conhecimentos e tecnologia tem sido fortemente valorizado na literatura (Kornecki et al., 2003; Rost, 2005).

É válido considerar também a alta rotatividade dos membros que participam do desenvolvimento de projetos de pesquisa. Conforme observado por Boldyreff et al. (2004), é comum, na prática, que as mesmas funcionalidades de um software desenvolvido em um projeto de pesquisa sejam implementadas diversas vezes por diferentes pessoas, havendo pouca ênfase na evolução do software. Assim, uma alternativa que pode ser considerada é melhorar a documentação dos projetos e disponibilizá-la aos membros que participam

¹http://www.finep.gov.br/fundos_setoriais/fundos_setoriais_ini.asp

²<http://www.cnpq.br>

³<http://www.cnpq.br/programasespeciais/coopint/index.htm>

do desenvolvimento nas diferentes fases da pesquisa e entre os membros das equipes geograficamente dispersas. A documentação pode incluir informações consideradas relevantes para o desenvolvimento e a continuidade dos projetos, por exemplo, as decisões que são tomadas e suas justificativas. Os conceitos de *design rationale* podem ser utilizados neste contexto com a finalidade de melhorar a documentação de projetos. De forma geral, *design rationale* se refere às informações relacionadas ao raciocínio empregado para resolver um problema e às descrições que justificam porque determinadas estruturas foram escolhidas em detrimento das demais alternativas (Gruber e Russel, 1991; Moran e Carroll, 1996).

Destacam-se, ainda, características que diferenciam um projeto de pesquisa em software do desenvolvimento de software. Em desenvolvimento de projetos de pesquisa (1) há o interesse no desenvolvimento de protótipos, que contribuam para a prova de conceitos; (2) o principal objetivo estratégico é a obtenção de resultados científicos. Em desenvolvimento de software, (1) há interesse no desenvolvimento de produtos; (2) os objetivos de negócio são fortemente considerados.

As características do desenvolvimento de projetos de pesquisa indicadas nesta seção representam elementos importantes que fundamentaram o desenvolvimento do trabalho apresentado nesta tese. Em resumo, foi observada a importância em considerar a qualidade dos artefatos produzidos e os aspectos de colaboração, distribuição e formação de parcerias na elaboração de um processo para desenvolvimento de projetos de pesquisa envolvendo software. Esses elementos foram considerados requisitos do processo proposto.

Nesta tese, desenvolvimento de projeto de pesquisa significa o processo de construção de conhecimento, que possui como metas principais confirmar ou refutar algum conhecimento pré-existente ou demonstrar novas idéias, por meio de provas de conceito.

1.2 Contexto e motivação

Conforme apresentado na seção anterior, é reconhecida a importância da qualidade nos resultados dos projetos de pesquisa que envolvem software e, portanto, tornam-se necessárias a definição e a melhoria dos processos para desenvolvimento de pesquisa, ainda que poucas iniciativas tenham sido apresentadas na literatura.

Buscando abordar os ciclos de vida para desenvolvimento de software, alguns estudos mais recentes estão sendo realizados com o objetivo de investigar a viabilidade de adaptar abordagens de desenvolvimento que já foram (ou que estão sendo) consolidadas para o contexto de desenvolvimento de pesquisa. Assim, por exemplo, Chirouze et al. (2005) propuseram uma adaptação do *eXtreme Programming* (XP) (Beck, 1999), denominada *eXtreme Researching* (XR), para oferecer apoio à pesquisa e desenvolvimento (P&D) realizada de forma distribuída. Em outro trabalho, Segal (2005) sugeriu a combinação

entre elementos de abordagens ágeis e tradicionais para desenvolvimento de software em projetos de pesquisa.

O fato de os trabalhos encontrados na literatura abordarem principalmente o ciclo de vida do software remete-nos a questionamentos sobre as demais atividades que são relevantes para o desenvolvimento de um projeto de pesquisa. Por exemplo, atividades de garantia de qualidade dos artefatos (não somente artefatos de software) foram pouco enfatizadas. Uma exceção é o trabalho desenvolvido por Walker (2003), em que foram consideradas atividades comuns do desenvolvimento de projetos de pesquisa. No entanto, somente um conjunto limitado de dez práticas base é sugerido e não houve a perspectiva de definição de um processo que incluísse artefatos de entrada, atividades e artefatos de saída.

Além disso, é possível notar que as características mencionadas na seção anterior, que determinam os requisitos de um processo para o desenvolvimento de pesquisas, foram tratados de forma isolada nas propostas apresentadas na literatura. Certamente, poderia ser considerado um domínio mais amplo, em que várias características fossem englobadas, tornando as propostas mais completas. É importante destacar, ainda, que a maioria das propostas apresentadas cobre apenas o aspecto de desenvolvimento de software nos projetos de pesquisa e não faz referência aos elementos do desenvolvimento de projetos de pesquisa. A motivação para o desenvolvimento deste trabalho está relacionada, portanto, à elaboração de uma proposta que cubra um conjunto de requisitos que foi identificado na literatura envolvendo atividades do desenvolvimento de pesquisa e atividades de desenvolvimento de software.

Ressalta-se também a motivação para o uso de *design rationale* neste trabalho. De acordo com Niazi et al. (2006), a falta de explicações detalhadas sobre a utilização dos processos na prática é uma das dificuldades na implantação de processos de software. Assim, observou-se que uma abordagem que foi utilizada na apresentação do CMMI (*Capability Maturity Model Integration*) (CMMI, 2006), e que contribui no sentido de oferecer informações adicionais sobre os processos, foi a apresentação de exemplos sugerindo como as atividades podem ser cumpridas e qual estrutura para os artefatos pode ser utilizada. Isso motivou a aplicação de conceitos de *design rationale* para a apresentação de exemplos, comentários e experiências que pudessem auxiliar o entendimento do processo proposto. Além disso, as potencialidades de *design rationale* foram exploradas para melhoria da documentação dos projetos desenvolvidos utilizando o processo apresentado nesta tese, com o objetivo de colaborar para a continuidade e a evolução dos projetos de desenvolvimento de pesquisas envolvendo software.

Este trabalho está inserido no contexto de definição de processos. O grupo de Engenharia de Software do ICMC-USP vem desenvolvendo pesquisas relacionadas à caracterização de processos para diferentes contextos (Barbosa, 2004; Reis, 2003b; Souza, 2005). Ressalta-se que a metodologia adotada por pesquisadores envolvidos nessa linha de

pesquisa contribuiu de forma significativa para direcionar o desenvolvimento do trabalho descrito nessa tese. De forma semelhante, espera-se que a metodologia adotada possa auxiliar a definição de outros processos, contribuindo para a continuidade e evolução da linha de pesquisa.

1.3 Objetivos e principal contribuição

O objetivo mais amplo deste trabalho foi identificar mecanismos que contribuíssem para o desenvolvimento de projetos de pesquisa na área de software. Sob essa perspectiva, dois elementos principais foram priorizados: (1) a definição de atividades do processo para a obtenção do conhecimento e (2) a evolução dos projetos de pesquisa. De forma mais específica, os principais objetivos deste trabalho foram:

- Definição de um processo para desenvolvimento de projetos de pesquisa envolvendo software;
- Estudo sobre *design rationale* (incluindo abordagens, utilização prática, restrições e dificuldades) e identificação de mecanismos que pudessem contribuir para melhorar a documentação dos projetos desenvolvidos.

Em relação à melhoria da documentação por meio de *design rationale*, o objetivo principal foi identificar um modelo de representação que contribuísse para o registro de informações sobre os projetos de pesquisa, considerando as características do contexto.

Considera-se que a principal contribuição e o principal elemento inovador deste trabalho estão relacionados ao modelo de representação de *design rationale* para o contexto de desenvolvimento de projetos de pesquisa que foi elaborado. O tema de definição de processos é bastante amplo, principalmente quando há interesse em considerar um conjunto extenso de requisitos na proposta. Assim, para que contribuições efetivas possam ser obtidas, torna-se fundamental a realização de investigações mais extensivas no sentido de validar o processo. Por outro lado, o estudo sobre a melhoria da documentação de projetos de pesquisa utilizando a abordagem de *design rationale* foi realizado considerando-se um foco bastante específico, que foi a atividade de engenharia de requisitos de software. Um estudo mais completo pôde ser realizado neste contexto: foi identificada a atividade de desenvolvimento de software cuja documentação poderia ser melhorada com o uso da abordagem (Paiva e Fortes, 2005), foram identificados requisitos para um modelo de *design rationale* para o contexto de desenvolvimento de pesquisas (Paiva et al., 2006a), foi elaborado um projeto para o modelo, tendo em vista a implementação de uma ferramenta (Freire, 2005), foi realizada a adaptação em uma ferramenta CASE para apoiar o uso do modelo (Paiva et al., 2006b) e foram realizados experimentos (Paiva et al., 2007).

1.4 Organização da tese

No Capítulo 2 é apresentada uma revisão bibliográfica sobre processos. São discutidas definições e características gerais de processos de software e, em seguida, são apresentados os principais elementos de processos mais específicos que foram relevantes para o desenvolvimento deste trabalho. Propostas apresentadas na literatura relacionadas a processos para o contexto de desenvolvimento de pesquisa também foram destacadas.

No Capítulo 3 é apresentado o conceito de *design rationale* e são discutidas as atividades de captura, representação e recuperação das informações. Os benefícios que podem ser obtidos com o uso de *design rationale* também são mencionados. Uma abordagem apresentada na literatura para a adaptação de *design rationale* para o contexto de desenvolvimento de pesquisas também é indicada.

No Capítulo 4 é descrito como o processo para desenvolvimento de projetos de pesquisa foi elaborado. Uma abordagem proposta por Humphrey (1995) foi utilizada como fundamento teórico e os resultados obtidos em relação à sua utilização são discutidos.

No Capítulo 5 é apresentado um resumo dos elementos que serviram como fundamento para a elaboração de um processo para desenvolvimento de projetos de pesquisa e os resultados obtidos em relação à definição do próprio processo. Além disso é apresentada uma discussão sobre a avaliação do mesmo.

No Capítulo 6 são apresentados resultados de investigações sobre a abordagem de *design rationale* realizadas com o objetivo de identificar mecanismos que auxiliassem a melhoria da documentação dos projetos de pesquisa. Um modelo de representação de *design rationale* foi proposto e os resultados obtidos com a realização de um experimento são discutidos.

No Capítulo 7 são discutidas as conclusões do trabalho bem como as perspectivas para trabalho futuro.

No Apêndice A são elencados projetos de pesquisa e suas atividades de desenvolvimento relacionadas. Esta análise foi útil durante a concepção do processo apresentado nesta tese, no sentido de auxiliar a compreensão das atividades cumpridas na prática, no contexto que está sendo estudado. No Apêndice B é apresentada a descrição do processo de desenvolvimento de projetos de pesquisa em software.

Fundamentos sobre Processos

2.1 Considerações iniciais

Neste capítulo é apresentada uma revisão bibliográfica sobre diferentes processos. Além de apresentar uma visão geral sobre processos de software, incluindo definições e características, foram priorizados processos que são usados em contextos específicos e que serviram como fundamento para a proposta do processo para desenvolvimento de projetos de pesquisa apresentada neste trabalho. Foram analisadas principalmente as características e as atividades referentes a esses processos e o estado da prática em termos de ferramentas automatizadas que promovam a implantação das atividades.

2.2 Visão geral sobre processos de software

Fuggetta (2000) apresentou uma definição para processos de software que tem sido utilizada por outros pesquisadores (Fiorini, 2001; González-Barahona e Robles, 2003) e que foi considerada neste trabalho. Segundo o autor, processos de software podem ser definidos como *“um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que são necessários para conceber, desenvolver, entregar e manter um produto de software”*. Além disso, o autor considera que o **ciclo de vida** do software define a estrutura e a filosofia segundo as quais o **processo** de software deve ser executado. As atividades e a forma como são organizadas, as ferramentas, os procedimentos operacionais, as políticas de desenvolvimento e as restrições são tratados em nível de processo. Assim, definir o ciclo de vida que será utilizado é uma atividade inicial importante, pois as pessoas envolvidas têm a oportunidade de começar a visualizar como será o desenvolvimento do software. No entanto, escolher um ciclo de vida

não é suficiente para que se consiga efetividade em seu controle e gerenciamento. É necessário tratar o desenvolvimento do projeto de software em nível de processo, observando políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos. Neste trabalho, a definição de um **processo** enfatiza os elementos apresentados por Fuggetta, considerando-se o domínio de desenvolvimento de projetos de pesquisa que incluem desenvolvimento de software.

Assim como Fuggetta, Humphrey e Kellner (1989) definiram processos de software de forma bastante genérica. Para eles, o conjunto de tarefas integradas que podem ser controladas, medidas e melhoradas e que estão envolvidas com o desenvolvimento e a manutenção de software constituem os processos de software.

A definição apresentada por Jaccheri e Conradi (1993) relaciona mais diretamente processos de software a ciclos de vida, pois eles sugerem que as seguintes atividades sejam executadas: análise de requisitos, projeto, codificação, testes, instalação e manutenção. As atividades estão relacionadas entre si, são compostas por outras atividades e operam sobre produtos de entrada para produzir produtos de saída.

De forma semelhante, Sommerville (2003) define processos de software como um conjunto de atividades de ciclo de vida, que se referem a especificação, desenvolvimento, validação e evolução do software. O autor acrescenta, em relação às definições anteriores, a possibilidade de definir processos de software de diversas formas. Para ele, os processos de software são descritos em diferentes níveis de detalhes, organizam as atividades de maneiras diversas e sugerem a elaboração de vários artefatos.

Para complementar a definição de processos de software é interessante observar as propriedades que processos no geral, e particularmente processos de software, devem possuir. Assim, de acordo com Pfleeger (2001), Madhavji (1991) e Kellner (1988), é fundamental que os processos:

- possuam razões que suportem suas definições;
- possuam muitas perspectivas, por exemplo, funcional, comportamental e organizacional;
- descrevam e integrem diferentes níveis de abstração;
- apresentem sintaxe e semântica definidas formalmente;
- sejam reutilizáveis;
- sejam compreensíveis, em abrangência e em detalhes.

A definição de processos de software é considerada importante, principalmente, por trazer disciplina à execução das diversas tarefas de desenvolvimento de software. Ao promover essa disciplina, obtém-se melhoria na capacidade dos processos, o que leva à

melhoria na qualidade e nos resultados dos projetos (Fuggetta, 2000; Krishman et al., 1999). Sob a mesma perspectiva, Kitchenham e Pfleeger (1996) indicaram a importância de processos de software para a obtenção de produtos de qualidade. Segundo as autoras, os processos são fundamentais para a produção de sistemas computacionais que estejam de acordo com suas especificações e para a melhoria da capacidade das organizações em produzir tais sistemas.

Tyrrell (2000) destacou ainda alguns objetivos primordiais relacionados à definição e utilização de processos de software:

- **alcançar efetividade** – um processo ajuda a produzir o produto certo, que está de acordo com o que o usuário necessita;
- **melhorar a manutenibilidade** – um processo favorece a documentação, que será fundamental quando mudanças forem necessárias;
- **fazer estimativas** – um processo ajuda a identificar os passos necessários para o desenvolvimento do sistema, facilitando as estimativas de prazo, tempo e recursos;
- **permitir replicação** – um processo documentado pode ser reusado em múltiplos projetos. É mais fácil e mais rápido adaptar um processo que já existe;
- **garantir a qualidade** – um processo ajuda a estabelecer uma ligação entre os requisitos do usuário e os artefatos desenvolvidos;
- **melhorar o próprio processo** – um processo documentado favorece a identificação de possibilidades de evoluções e melhorias no processo;
- **entender o projeto** – um processo documentado favorece o entendimento das atividades executadas para o desenvolvimento de um projeto por gerentes, desenvolvedores e clientes.

Tyrrell discutiu também a importância em definir processos que seres humanos, e não máquinas, sejam capazes de executar. Segundo ele, é fundamental fornecer um conjunto de guias que indiquem quais atividades devem ser executadas e quais documentos devem ser produzidos. O objetivo é tornar a execução do processo mais fácil e diminuir a sobrecarga para analistas e desenvolvedores. Discussões semelhantes foram apresentadas por Humphrey (1995) para a definição de processos.

A definição e os objetivos dos processos de software são bastante abrangentes e, portanto, podem ser observados sob diferentes perspectivas, de acordo com os contextos nos quais são adotados. A seguir, como um resultado da revisão bibliográfica, é apresentada uma visão geral da evolução dos processos de software.

2.3 Desenvolvimento distribuído de projetos

Uma importante evolução da Engenharia de Software (e uma tendência para os próximos anos) está associada a forma como suas atividades passaram a ser executadas. Inicialmente havia um enfoque individual, ou seja, cada atividade era desenvolvida por uma pessoa da equipe de desenvolvimento ou por um número pequeno de pessoas, em geral, fisicamente próximas. Nos últimos anos, com o aumento da complexidade dos projetos de software tornou-se imprescindível a interação e a colaboração entre membros de diversas áreas de conhecimento, que muitas vezes não se encontram em um mesmo local de trabalho, levando à distribuição de tarefas (Arnold, 1994; Hawthorne e Perry, 2005).

De acordo com Maurer e Kaiser (1998) e Perpich et al. (1997) diversos elementos críticos precisam ser considerados em processos utilizados para o desenvolvimento distribuído de projetos, por exemplo, comunicação entre os membros, acesso ao conhecimento, desenvolvimento de atividades cooperativas e registro de experiências adquiridas para posterior reutilização.

Sob a mesma perspectiva, pesquisadores que estão investigando e experimentando processos de software no desenvolvimento distribuído de projetos indicaram as principais atividades que os compõem. Maidantchik (1999) identificou atividades que devem ser acrescentadas à norma ISO/IEC 12207 com o objetivo de definir um processo padrão que possa ser utilizado por equipes de trabalho heterogêneas, geograficamente dispersas e culturalmente diferentes para o desenvolvimento de software. Komi-Sirvio e Tihinen (2005) realizaram um *survey* com analistas e desenvolvedores que participaram do desenvolvimento distribuído de projetos com o objetivo de compreender as características do desenvolvimento distribuído, os problemas associados e possíveis soluções. Battin et al. (1994) descreveram suas experiências com o desenvolvimento distribuído de projetos na Motorola e indicaram as principais características de um projeto distribuído e os problemas que geralmente são enfrentados pelas equipes e as possíveis soluções. Prikladnicki et al. (2004) e Herbsleb et al. (2005) apresentaram lições que foram aprendidas a partir do desenvolvimento de projetos distribuídos. De forma geral, os pesquisadores destacaram a importância das atividades de gerenciamento de configuração, comunicação, gerenciamento e coordenação de projetos e definição e estabelecimento da infra-estrutura. Komi-Sirvio e Tihinen (2005), Prikladnicki et al. (2004) e Herbsleb et al. (2005) mencionaram também a importância do registro de experiências e decisões de projeto (que é um dos focos deste trabalho) para auxiliar na divulgação de informações de interesse das equipes.

Nota-se que há uma demanda não somente em relação a processos para o desenvolvimento distribuído de projetos, mas também em relação a ambientes e ferramentas que possam apoiar o cumprimento desses processos. A plataforma GENESIS (*GEneralised*

eNvironment for procEsS management in cooperatIve Software engineering) (Aversano et al., 2004), por exemplo, está sendo desenvolvida como um projeto de pesquisa que tem como objetivo projetar e desenvolver um sistema para apoiar processos de Engenharia de Software em um ambiente distribuído. Dentre outras funcionalidades, o sistema oferece recursos para apoio à cooperação, à coordenação dos processos, à modelagem e decomposição dos processos em sub-processos (de forma que possam ser executados por diferentes equipes) e ao gerenciamento de recursos, de artefatos e de dados.

De forma semelhante, foi proposto por Kotting e Maurer (1999) um sistema para gerenciamento de *workflow*, denominado MILOS, que utiliza recursos da internet para apoiar a coordenação dinâmica de equipes de desenvolvimento de software. Uma das principais características do sistema é o suporte ao desenvolvimento de projetos de software ágeis e distribuídos (Bowen e Maurer, 2002).

Kammer et al. (1998), Kochut et al. (2003) e Cugola et al. (2001) também propuseram ambientes para apoiar o desenvolvimento de projetos distribuídos. De forma geral, as principais funcionalidades implementadas estão relacionadas à definição e especificação de artefatos, atividades e recursos do processo, coordenação do processo, gerenciamento de *workflow* (criação, execução e monitoramento do *workflow*) e apoio à comunicação.

2.4 Desenvolvimento colaborativo de projetos

Conforme apresentado na seção 2.3, um dos principais fatores que motivaram a mudança de paradigma em relação à forma de desenvolvimento de projetos de software (de uma abordagem individual para uma abordagem voltada para equipes) foi o aumento da complexidade desses projetos e a necessidade de colaboração entre pessoas de diferentes áreas de conhecimento. O advento da internet e os avanços tecnológicos em termos de redes de comunicação foram fatores fundamentais que possibilitaram que a mudança ocorresse.

O desenvolvimento colaborativo se refere ao envolvimento de diversos grupos de pessoas trabalhando conjuntamente no desenvolvimento de um projeto, para benefício de todos os participantes (Uwadia et al., 2006). O TSP (*Team Software Process*) foi proposto por Humphrey (1999) como uma evolução do PSP, com a finalidade de servir a grupos de pessoas que tenham por objetivo desenvolver software de qualidade. De forma geral, o PSP (*Personal Software Process*) (Humphrey, 1995) foi projetado para auxiliar estudantes e engenheiros de software a organizar e planejar seus trabalhos, medir seus desempenhos, gerenciar a qualidade de software e analisar e melhorar seus processos. No TSP são apresentados *guidelines*, atividades, ferramentas, métodos e técnicas para desenvolver software por equipes. O TSP é baseado em um modelo incremental, sendo que as atividades que compõem cada ciclo de desenvolvimento são executadas de forma seqüencial.

Os participantes da equipe de desenvolvimento são organizados de forma que cada desenvolvedor desempenhe um ou dois papéis gerenciais bem definidos. Os papéis suportados pelo processo são: gerente de desenvolvimento, de planejamento, de qualidade, de processo, de suporte e líder da equipe. O planejamento e o controle rigoroso de tamanhos, esforços, prazos e defeitos apresentados no PSP também foram considerados no TSP. Os principais processos enfatizados no TSP são gestão de requisitos, planejamento e controle de projetos, garantia da qualidade e gerenciamento de configuração.

De acordo com Hilburn (2000), o TSP é excelente para auxiliar estudantes no aprendizado da prática profissional de Engenharia de Software e, sobretudo, como trabalhar em uma equipe para o desenvolvimento de software. Como desvantagem, o autor menciona que a execução do processo é bastante onerosa em termos de tempo gasto para realizar as atividades propostas e coletar os dados sugeridos. Além disso, como indicado por Pressman (2005), a aplicação do PSP e do TSP exige treinamento extensivo e muita disciplina das pessoas envolvidas no desenvolvimento dos projetos. Filho (2001b) observou também, a partir de experiências práticas, que a aplicação dos processos é bem sucedida quando os membros das equipes possuem excelente conhecimento em Engenharia de Software e possuem motivação para empregar processos de software em nível avançado.

Nota-se, portanto, que o requisito de que as pessoas envolvidas possuam conhecimentos avançados em Engenharia de Software desfavorece a aplicação direta do TSP no contexto de desenvolvimento de projetos de pesquisa. É comum que os estudantes e pesquisadores não possuam experiência na realização das diversas atividades, que envolvem, por exemplo, controle de qualidade e gerenciamento do projeto. No processo proposto neste trabalho, tendo em vista o auxílio à execução e evolução de projetos de pesquisa em software, considera-se que os participantes podem ser *aprendizes* e que a participação em um projeto de pesquisa pode ser, inclusive, uma de suas primeiras experiências práticas.

Pesquisadores que estão investigando e experimentando processos de software no desenvolvimento colaborativo de projetos indicaram as principais atividades que devem ser consideradas. Paasvaara e Lassenius (2004) realizaram entrevistas com participantes de oito projetos desenvolvidos de forma colaborativa em que foi solicitado que eles indicassem as atividades que consideraram mais importantes para o sucesso dos projetos; Augustin et al. (2002) discutiram as práticas de desenvolvimento de software colaborativo que eles consideraram mais importantes de acordo com suas experiências no desenvolvimento de projetos colaborativos; Ebert e Neve (2001) apresentaram, considerando suas experiências em uma empresa de telecomunicações, práticas que julgaram relevantes e que consideraram que devam ser cumpridas com efetividade para que o desenvolvimento de projetos colaborativos seja bem sucedido; Pinheiro et al. (2002) e Araújo e Borges (2007) propuseram ferramentas que auxiliam o desenvolvimento colaborativo de projetos. De forma geral, os autores destacaram a prática de definição de papéis e do apoio à comunicação entre os membros das equipes. Notou-se, ainda, que os autores mencionaram

usando diferentes nomenclaturas, a importância da *percepção* do trabalho desenvolvido pelos membros dos grupos. É interessante notar que as práticas indicadas pelos autores são semelhantes e estão bastante relacionados às práticas para o desenvolvimento distribuído de projetos. De fato, conforme indicado por Cook et al. (2004), para que a Engenharia de Software colaborativa possa ser colocada em prática de fato, áreas de pesquisa como visualização de software, interação usuário-computador e desenvolvimento distribuído devem contribuir efetivamente, ou seja, a interseção entre essas áreas forma a base do desenvolvimento colaborativo.

Diversos ambientes virtuais colaborativos foram propostos nos últimos anos, dentre eles os sistemas DIVE (Hagsand, 1996), Manufaktur (Buscher et al., 2000) e CARAMBA (Dustdar, 2004). As funcionalidades implementadas são bastante semelhantes e se referem basicamente ao compartilhamento da área de trabalho, gerenciamento do projeto, gerenciamento do conhecimento adquirido, gerenciamento de documentação e comunicação.

2.5 Processo de software livre

Software livre pode ser definido como qualquer software cuja licença garanta aos usuários liberdade para usar, copiar, distribuir, estudar, modificar e melhorar o software (Open Source Initiative, 2006; The Free Software Foundation, 2006). Esta definição, embora seja adotada por entidades importantes que lidam com o desenvolvimento de software livre, enfatiza o produto final gerado. É importante destacar também determinadas características com as quais o desenvolvimento de software livre está associado (Raymond, 2001):

1. Desenvolvimento descentralizado via internet: quando há mais de uma pessoa envolvida, o desenvolvimento é realizado de forma colaborativa, usando a internet como meio de comunicação (sites *web* e FTP, repositórios de versões e correio eletrônico);
2. Participação dos usuários: é comum a formação de um grupo de usuários finais que se comunicam com alguma regularidade com os desenvolvedores e entre si, comunicando problemas e trocando experiências sobre o uso do software;
3. Interesse pessoal do autor: geralmente o autor é um usuário do software e, portanto, tem motivação pessoal na sua criação e manutenção.

Os modelos utilizados para desenvolvimento de software livre apresentam características diferentes em relação aos modelos tradicionais de Engenharia de Software (Scacchi et al., 2006). Em projetos de software livre, os artefatos produzidos são disponibilizados em repositórios de controle de versões e não há um regime formal para o gerenciamento dos projetos, dos custos e dos prazos. Em um dos principais trabalhos

realizados nesse sentido, Raymond (1999) identificou dois modelos distintos para software livre: (1) o modelo Catedral, em que os projetos de software livre são desenvolvidos por grupos fechados, em longos ciclos de desenvolvimento (e conseqüentemente com muita demora para a liberação das versões) e com pequena abertura à participação externa e (2) o modelo Bazar, que descreve projetos desenvolvidos de forma mais transparente, abertos à participação de desenvolvedores que tenham interesse, com tempo de desenvolvimento curto e alta qualidade.

Algumas pesquisas têm sido realizadas com o objetivo de entender e caracterizar o processo de desenvolvimento de software livre (Feller e Fitzgerald, 2000; O'Reilly, 1999). De acordo com Reis (2003b), podem ser observadas muitas variações nos processos utilizados para o desenvolvimento dos projetos e, portanto, devem ser evitadas generalizações.

Apesar de não haver um processo estabelecido, que sirva como referência para os desenvolvedores de software livre (a modelagem do processo de software livre e a simulação sobre como ele opera ainda é um desafio de pesquisa), pode-se notar que algumas atividades principais geralmente são enfatizadas. Assim, de acordo com Reis (2003b) e Zhao e Elbaum (2003):

- há pouca ênfase nas fases de especificação de requisitos e de projeto e muitos esforços são direcionados à fase de codificação;
- as atividades de gerenciamento de configuração de software e rastreamento de *bugs* são realizadas em grande parte dos projetos;
- os testes são realizados intensivamente pelos usuários, que colaboram indicando *bugs* e sugerindo melhorias;
- a documentação enfatiza principalmente as funcionalidades do software. São realizadas atividades de documentação simples, como elaboração de listas contendo as tarefas pendentes (*TODO lists*) e elaboração de um guia de instalação;
- revisões de código são realizadas com freqüência.

Ferramentas

Nota-se que ferramentas de controle de versões, listas de correio eletrônico e outras ferramentas de comunicação são amplamente utilizadas na comunidade de software livre. Uma parcela menor, porém significativa, dos participantes utiliza sistemas de acompanhamento de alterações. Assim, observando-se as ferramentas mais comumente utilizadas, é possível identificar os processos mais valorizados. Pode-se observar que a execução de atividades que oferecem suporte à comunicação entre os membros que

participam dos projetos e de atividades relativas ao gerenciamento de configuração é um dos fatores mais importantes para o sucesso dos projetos.

Para coordenar o trabalho que realizam, os membros da comunidade de um projeto utilizam a internet e ferramentas simples amplamente disponíveis: correio eletrônico, páginas *web*, listas de discussão e ferramentas de desenvolvimento de software. Em geral, os membros dos projetos tendem a usar estes veículos para comunicar experiências, problemas e solicitações. O uso intensivo dessas ferramentas apóia o desenvolvimento de projetos distribuídos, como ocorre no contexto da maioria dos projetos de software livre (Reis, 2003b).

É possível observar que existem características em comum em relação ao desenvolvimento de software livre e desenvolvimento de projetos de pesquisa. De acordo com Alpern et al. (2005), a experiência obtida com o desenvolvimento de projetos de software livre deve ser observada por pesquisadores interessados em desenvolver pesquisa colaborativa. Conforme apresentado por Bezroukov (1999) e Ambati e Kishore (2004) há muitas similaridades entre os processos de software livre e os processos acadêmicos. Por exemplo, o desenvolvimento distribuído e colaborativo pode ser considerado uma característica importante em ambos os processos. Bezroukov (1999) e Ambati e Kishore (2004) sugeriram que dificuldades comuns no desenvolvimento de software em ambiente de pesquisa, como a comunicação e a coordenação de projetos, sejam resolvidas utilizando-se práticas e ferramentas amplamente utilizadas na comunidade de software livre, como ferramentas de gerenciamento de mudanças e de versões. De forma similar, eles observam que atividades comumente executadas no desenvolvimento de projetos de pesquisa podem ser adotadas por desenvolvedores de software livre com o objetivo de solucionar problemas que têm sido percebidos por eles. Por exemplo, na comunidade de software livre há grande preocupação em relação ao nível de conhecimento entre especialistas e novatos. Em ambiente acadêmico, há disponibilidade e interesse em treinar desenvolvedores novatos, mas isso nem sempre ocorre no desenvolvimento de software livre. Ambati e Kishore sugeriram, portanto, adotar a idéia de “comunidade de assistentes de ensino” no contexto de software livre, da forma como ocorre em ambiente acadêmico.

2.6 Métodos ágeis

Em 2001, um grupo de profissionais experientes em desenvolvimento de software se reuniu com o objetivo de discutir alternativas para os métodos tradicionais de desenvolvimento de software, que são fortemente baseadas em documentação. Como resultado dessa reunião, foi criado o Manifesto para o Desenvolvimento Ágil de Software (ou Manifesto Ágil) (Beck et al., 2001), para o qual foram defendidos os seguintes valores:

- pessoas e interações são mais importantes que processos e ferramentas;

- software funcional é mais importante que documentação detalhada;
- colaboração do cliente é mais importante que negociação;
- resposta a mudanças é mais importante que cumprimento de um plano.

Além disso, foram identificados doze princípios que regem os métodos ágeis de desenvolvimento de software: (1) entregar versões continuamente e o mais cedo possível; (2) permitir mudanças de requisitos em qualquer fase do desenvolvimento; (3) entregar versões do software, que funcionam adequadamente, em curtos períodos de tempo (de algumas semanas a alguns meses); (4) desenvolver o projeto com a participação de clientes; (5) fornecer infra-estrutura necessária para que os clientes se sintam motivados e desempenhem o trabalho esperado; (6) incentivar a comunicação face a face; (7) alcançar a medida primária de progresso, que é o software operacional; (8) manter harmonia entre clientes, desenvolvedores e usuários; (9) preocupar-se com a qualidade técnica e com a execução de bons projetos; (10) projetar com simplicidade; (11) permitir que equipes da própria organização participem da definição da arquitetura, requisitos e projeto do sistema e (12) permitir que as equipes, em intervalos regulares, expressem como podem se tornar mais eficientes.

De acordo com Highsmith e Cockburn (2001), a novidade dos métodos ágeis não são as práticas propostas, mas sim o reconhecimento de que *pessoas* são os principais elementos para o sucesso dos projetos. Os autores destacam também que, de forma geral, métodos ágeis são projetados com o objetivo de (1) auxiliar a produção das primeiras versões do software em semanas, possibilitando a obtenção de *feedback* rápido, (2) produzir soluções simples, (3) auxiliar a melhoria de qualidade continuamente, tornando a próxima iteração menos custosa e (4) promover a execução de testes constantemente, de forma que defeitos sejam identificados o quanto antes, a custos mais baixos.

Williams e Cockburn (2003) indicaram as circunstâncias nas quais métodos ágeis devem ser usados: o sistema a ser desenvolvido não é de missão crítica, os requisitos mudam com frequência, a equipe de desenvolvimento é pequena e situada na mesma localidade e clientes, usuários e especialistas no domínio do problema estão dispostos a colaborar no desenvolvimento. Conforme apresentado por Kircher et al. (2001) e por Yap (2005) foi observada, mais recentemente, a possibilidade de que métodos ágeis sejam utilizados também por equipes distribuídas.

É possível observar semelhanças entre as características do ambiente de desenvolvimento de projetos de pesquisa para o qual o processo apresentado nesta tese está sendo proposto e as características do ambiente no qual métodos ágeis podem ser utilizados. Pretende-se que o processo seja utilizado por equipes pequenas e geograficamente dispersas. Outra característica em comum se refere às mudanças frequentes dos requisitos. Conforme apresentado por Segal (2005) e Huljenic et al. (2005), os requisitos de um

protótipo resultante de um projeto de pesquisa são definidos gradualmente, à medida que o projeto é desenvolvido. Esses requisitos mudam com frequência, de acordo com os resultados e o progresso do projeto.

As atividades que compõem os métodos ágeis variam de acordo com as diferentes propostas apresentadas na literatura, tais como, *eXtreme Programming* (XP) (Beck, 1999), SCRUM (Schwaber e Beedle, 2002) e *Feature Driven Development* (FDD) (Palmer e Felsing, 2002). O método XP tem sido utilizado no contexto do desenvolvimento de projetos de pesquisa (Bellotti et al., 2002; Schneider e Johnston, 2003) e, portanto, uma breve descrição do método, baseada em Abrahamsson et al. (2002), é apresentada a seguir.

O ciclo de vida de XP consiste de cinco fases: exploração, planejamento, iterações para produção de versões, produção, manutenção e finalização (Beck, 1999). Na fase de exploração, os clientes escrevem as histórias que eles querem que sejam incluídas na primeira versão. Ao mesmo tempo, a equipe de projeto familiariza-se com as ferramentas, tecnologias e práticas que serão usadas no projeto. São exploradas possibilidades de arquitetura para o sistema. Na fase de planejamento, é definida uma ordem para as histórias e é feito um acordo sobre o conteúdo da primeira *release*. Os programadores estimam esforço e prazo requeridos para cada história. Na fase de iterações, o cronograma definido na fase anterior é dividido em um número de iterações. Na primeira iteração é criada a arquitetura de todo o sistema. Os clientes decidem as histórias que serão selecionadas para cada iteração. Os testes funcionais são executados ao final de cada iteração. Ao final da última iteração, o sistema está pronto para a fase de produção, em que são realizados novos testes e é avaliado o desempenho do sistema. Depois que a primeira *release* é entregue ao cliente, a equipe de projeto deve manter o sistema em produção. Assim, na fase de manutenção, o esforço é direcionado a tarefas de suporte ao cliente. A fase de finalização ocorre quando o cliente não possui mais histórias a serem implementadas. Este é o momento em que a documentação do sistema é escrita. A finalização pode ocorrer também se o sistema não está produzindo os resultados esperados ou torna-se muito caro.

Em relação à escolha de um método ágil ou tradicional para o desenvolvimento de um projeto, conforme apresentado por Hawrysh e Ruprecht (2000), é importante considerar que não existe uma opção que possa ser adotada em um amplo espectro de diferentes tipos de projetos. Deve-se, ao contrário, identificar a natureza específica e as principais características do projeto no início do desenvolvimento e, então, selecionar o melhor método aplicável. Sob a mesma perspectiva, McCauley (2001) reforçou a importância em conjugar práticas de métodos ágeis e tradicionais, em que a documentação é fortemente recomendada, pois, segundo ele, não há um modelo de desenvolvimento de software que possa ser utilizado universalmente. De forma semelhante, em relação ao desenvolvimento de projetos de pesquisa, Segal (2005) recomendou, a partir de suas experiências práticas, que sejam combinadas práticas de métodos ágeis e tradicionais. Segundo o autor,

esta combinação poderia ajudar a solucionar problemas relacionados à documentação e comunicação, que têm sido freqüentemente experimentados pela comunidade científica.

2.7 Qualidade de processos de software

Modelos de qualidade de software que auxiliam a melhoria de processo de software foram propostos com o objetivo de contribuir para a definição e o aprimoramento de processos de software, devido à complexidade envolvida na implementação das atividades relacionadas. A melhoria de processos significa compreender os processos existentes e modificá-los, a fim de melhorar a qualidade do produto e/ou reduzir os custos e o tempo de desenvolvimento (Sommerville, 2003). Para a apresentação do processo padrão foi utilizada a estrutura indicada na norma ISO/IEC 12207, em que os processos são divididos em processos fundamentais, processos de apoio e processos organizacionais. Para a especialização do processo padrão foi utilizada a estrutura em níveis do CMMI. Uma breve descrição desses modelos de qualidade de software é apresentada a seguir.

Norma ISO/IEC 12207

A norma ISO/IEC 12207 tem por objetivo auxiliar os envolvidos na produção de software a definir seus papéis, por meio de processos bem definidos, possibilitando às organizações que a utilizam um melhor entendimento das atividades a serem executadas nas operações que envolvem software (Rocha et al., 2001b). Os processos que foram incluídos na norma ISO/IEC 12207 são apresentados na Figura 2.1.

Nos **processos fundamentais** estão incluídos os processos de aquisição, de fornecimento, de desenvolvimento, de operação e de manutenção. No processo de aquisição, ao ser identificada a necessidade de aquisição de um sistema, de um produto ou de um serviço de software, é preparado e emitido o pedido de proposta e são definidos os critérios de aceitação ou rejeição do item adquirido. No processo de fornecimento são realizadas a preparação de uma resposta a um adquirente, a assinatura de um contrato, a determinação dos procedimentos e recursos necessários para gerenciar e garantir o projeto e a entrega do sistema. No processo de desenvolvimento são definidas as atividades do ciclo de vida de desenvolvimento do software. No processo de operação são definidas as atividades que serão executadas para orientação ao usuário e envio de informações para o processo de manutenção do produto. No processo de manutenção são definidas as atividades que o mantenedor do software deverá executar.

Nos **processos de apoio** estão incluídos o processo de documentação, de gerência de configuração, de garantia de qualidade, de verificação, de validação, de revisão conjunta, de auditoria e de resolução de problemas. No processo de documentação são definidas as atividades para planejar, projetar, desenvolver, produzir, editar, distribuir e manter

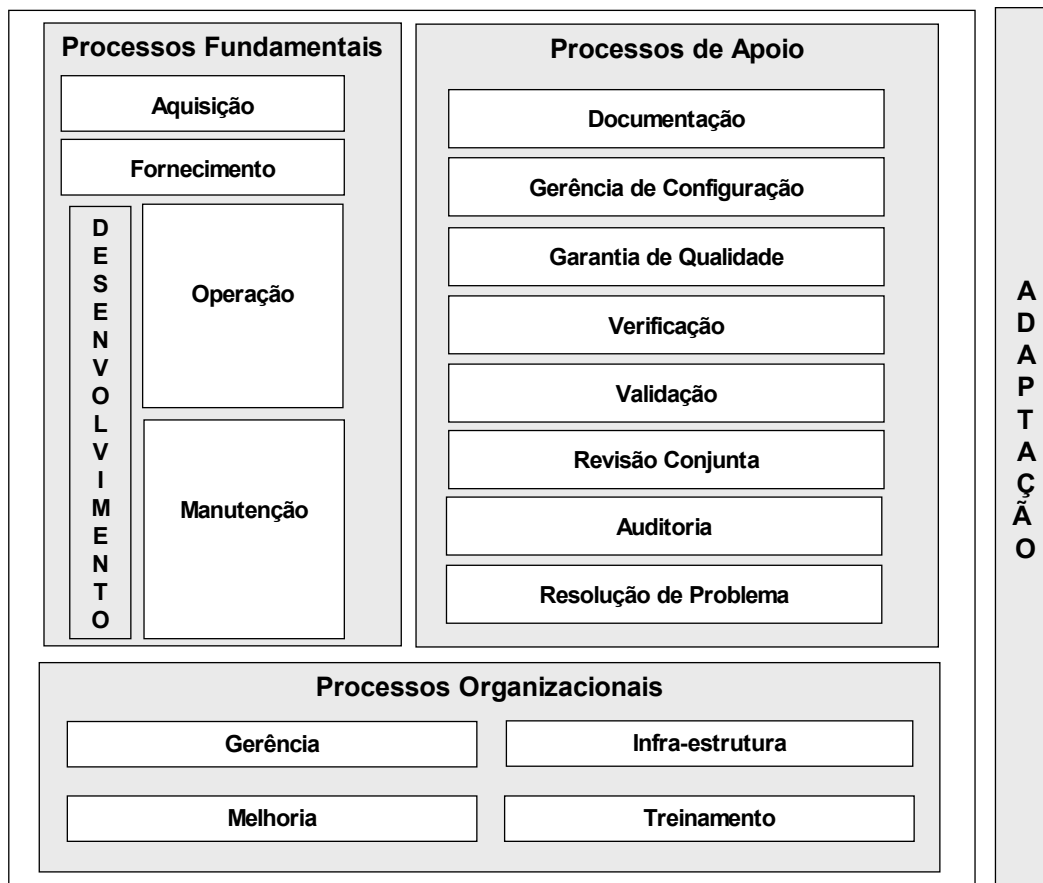


Figura 2.1: Estrutura da Norma ISO/IEC 12207 (1995)

os documentos. No processo de gerência de configuração, os itens de software são identificados, as modificações, o armazenamento, a manipulação e a distribuição dos itens de software são controlados e as versões são gerenciadas. No processo de garantia da qualidade, os processos e produtos de software são avaliados em relação aos requisitos e planos. No processo de verificação os produtos de software são avaliados em relação à corretude técnica. No processo de validação é avaliado se o produto de software cumpre com o objetivo de uso para o qual foi construído. No processo de revisão, as atividades do processo e os artefatos resultantes são avaliados. No processo de auditoria, a adequação do produto aos requisitos, aos planos e ao contrato é avaliada. No processo de resolução de problemas é definido um processo para analisar e resolver problemas de qualquer natureza, detectados durante o desenvolvimento, a operação, a manutenção ou a execução de outros processos.

Nos **processos organizacionais** estão incluídos o processo de gerência, o processo de infra-estrutura, o processo de melhoria e o processo de treinamento. No processo de gerência, são definidas as atividades que devem ser executadas para gerenciamento do produto, do projeto e das tarefas relacionadas ao desenvolvimento do software, tais como, aquisição, fornecimento, desenvolvimento, operação, manutenção e apoio. No processo

de infra-estrutura são definidas as atividades para estabelecer e manter a infra-estrutura que provê apoio ao ciclo de vida do software. No processo de melhoria são definidas as atividades básicas que devem ser desempenhadas para estabelecer, medir, avaliar, controlar e melhorar o ciclo de vida do software. No processo de treinamento são definidas as atividades para prover treinamento e manter o pessoal treinado.

No **processo de adaptação** são definidas as atividades necessárias para adaptar a norma a uma organização ou a projetos específicos. As atividades de adaptação devem considerar, por exemplo, estratégias, procedimentos, políticas e culturas organizacionais, tamanho, criticalidade e tipo de sistema, modelo de ciclo de vida do projeto, características do sistema, riscos, custos e pessoal envolvidos.

Modelo CMMI

O modelo CMMI (*Capability Maturity Model Integration*) foi proposto como uma evolução dos diversos modelos CMM (*Capability Maturity Model*) (Paulk et al., 1995) apresentados pelo SEI (*Software Engineering Institute*) a partir da década de 1990. O resumo apresentado a seguir foi adaptado de Koscianski e Soares (2006).

O objetivo do CMMI é servir de guia para melhoria de processos na organização e também da habilidade dos profissionais em gerenciar o desenvolvimento, a aquisição e a manutenção de produtos ou serviços. Assim como no caso do CMM, espera-se que, com o uso do CMMI, a organização seja mais eficiente, respeitando seus próprios prazos e construindo software com menos erros.

Há quatro áreas de conhecimento (ou disciplinas) presentes no modelo CMMI: (1) engenharia de sistemas, (2) engenharia de software, (3) desenvolvimento e integração de produtos e processos e (4) fontes de aquisição. A primeira delas tem por objetivo a obtenção bem sucedida de sistemas, envolvendo software ou não. A engenharia de software tem o intuito de disciplinar a produção de software. O desenvolvimento e integração do produto e do processo é uma abordagem sistemática que utiliza a colaboração dos participantes do projeto para melhor satisfazer as expectativas e requisitos dos clientes. A disciplina de fontes de aquisição atua na aquisição de produtos ou serviços para o desenvolvimento dos projetos, pois, à medida que os esforços de desenvolvimento aumentam, os projetos podem precisar de fornecedores que realizem funções específicas ou prestem manutenções nos artefatos gerados.

Há duas representações para o CMMI, denominadas representação por estágios e representação contínua. A representação por estágios segue a mesma estrutura utilizada no CMM, ou seja, as áreas de processo são organizadas em níveis de maturidade para guiar a melhoria de processos. Os níveis representam um caminho de melhoria de processo para toda a organização. Os níveis de maturidade sugerem uma ordem para a melhoria dos processos. Para cada nível de maturidade existem áreas de processo. Em cada área de processo há objetivos e práticas genéricas e específicas.

Na representação por estágios existem cinco níveis de maturidade: (1) inicial, (2) gerenciado, (3) definido, (4) gerenciado quantitativamente e (5) em otimização. Organizações que estão no nível inicial não possuem um ambiente estável de desenvolvimento de software; no nível gerenciado, desenvolvem projetos cujos requisitos são gerenciados e os processos são planejados, medidos e controlados; no nível definido, os processos são padronizados e há maior consistência nos produtos gerados pela organização; no nível gerenciado quantitativamente, os processos são controlados usando métodos estatísticos; no nível em otimização, os processos são continuamente melhorados com base em um entendimento quantitativo das causas comuns de alterações de desempenho.

Na representação contínua, cada processo da organização é avaliado separadamente. Existem seis níveis de capacidade: (0) incompleto, (1) realizado, (2) gerenciado, (3) definido, (4) gerenciado quantitativamente e (5) em otimização. O nível incompleto corresponde ao não cumprimento de um processo para o desenvolvimento dos projetos. No nível realizado, há um processo que possui entradas e saídas bem definidas. No nível gerenciado, cada processo cumpre com todos os requisitos do nível realizado e, além disso, é planejado e executado de acordo com uma política determinada. No nível definido, os processos da organização são padronizados. No nível gerenciado quantitativamente, os processos são controlados usando métodos estatísticos. No nível em otimização, o foco está na melhoria contínua de desempenho dos processos da organização.

2.8 Processos e ambientes para desenvolvimento de projetos de pesquisa

Conforme apresentado por Ambati e Kishore (2004), as características especiais do desenvolvimento de projetos de pesquisa, tais como a prototipação rápida e a alta rotatividade dos participantes representam elementos fundamentais que impõem uma revisão das práticas de Engenharia de Software para atender às necessidades do contexto de desenvolvimento de pesquisas. Em geral, espera-se que um processo executado em ambiente de pesquisa valorize, principalmente, a comunicação entre os membros dos projetos, as atividades de prototipação e de gerenciamento de tarefas, a definição de cronograma, o planejamento de recursos e as atividades de qualidade de software, envolvendo documentação, gerenciamento de configurações e testes (Filho, 2001b; Robillard e Robillard, 1998; Segal, 2005).

Nos últimos anos, foram apresentadas propostas de processos com o objetivo de atender ao desenvolvimento de projetos de pesquisa. Embora estejam sendo obtidos resultados importantes, na prática, ainda é comum que os protótipos desenvolvidos sejam re-escritos por diferentes pessoas, várias vezes, devido às dificuldades de manutenção e reuso, acarretando em prejuízos à evolução de pesquisas científicas (Boldyreff et al., 2004).

Nas seções seguintes são descritas as propostas que foram encontradas na literatura sobre processos para o desenvolvimento de projetos de pesquisa, considerando-se seus aspectos positivos e negativos. Na Seção 5.6.2 é apresentada uma análise comparativa entre as propostas e o processo apresentado neste trabalho.

2.8.1 *eXtreme Researching*

Chirouze et al. (2005) propuseram uma adaptação do método XP, denominada *eXtreme Researching* (XR), para desenvolvimento de projetos de pesquisa. A proposta adota e estende as principais práticas de XP e mostra como podem ser aplicadas em projetos distribuídos que envolvem o desenvolvimento de pesquisas (foi considerado como requisito fundamental para a elaboração da proposta que o desenvolvimento de pesquisas ocorre de forma distribuída). Os autores consideram que um dos principais desafios, neste contexto, é propor um método que valorize a disciplina em termos de atividades e artefatos e, ao mesmo tempo, a criatividade dos pesquisadores.

Foi adotado um método ágil como base para a proposta devido a suas principais propriedades, ou seja, são orientadas a pessoas, ao invés de serem orientadas a processos; são adaptativas, ao invés de serem preditivas ou impositivas e consideram que os requisitos do software são descobertos à medida que o projeto é desenvolvido. Os autores consideraram que há uma relação entre estas propriedades e o ambiente de desenvolvimento de projetos de pesquisa.

Para a elaboração do método XR, foi considerado, além do método XP, o trabalho desenvolvido por Kircher et al. (2001), que propôs uma abordagem para execução das práticas originais de XP de forma distribuída. Um dos principais elementos tratados pelos autores foi a comunicação entre os membros dos projetos, tendo sido proposto o uso de ferramentas de *email*, teleconferência e videoconferência. Assim, ao adotar os resultados do trabalho de Kircher et al., a aplicação do método XR fica restrita a grupos que possuam a infra-estrutura e os recursos necessários para o estabelecimento da comunicação de acordo com a proposta (na prática, nem todas as ferramentas sugeridas são amplamente difundidas). Na Tabela 2.1 são apresentadas as práticas de XR que foram propostas.

Foi apresentado também um portal *web*, denominado XPWeb¹, que oferece apoio ao desenvolvimento de projetos utilizando XP e disponibiliza os repositórios necessários para o desenvolvimento de projetos de acordo com o método XR. O principal objetivo do portal é fornecer um conjunto de ferramentas que promova o compartilhamento de conhecimento em um ambiente distribuído e seja útil para o desenvolvimento de provas de conceito. Outra característica importante é que o portal possui interfaces para vários sistemas,

¹<http://sourceforge.net/projects/xpweb>

Tabela 2.1: Práticas do método XR (Chirouze et al., 2005)

Atividades e Características	Descrição e Informações Gerais
Integração freqüente	Realizada por meio do uso de um repositório <i>online</i> , sempre disponível para membros dos grupos. Não há uma fase de integração específica. Pesquisadores desenvolvendo uma prova de conceito esperam obter <i>feedback</i> rápido do código para refinar uma primeira idéia
Programação por pares remotos	Executada a partir do compartilhamento da área de trabalho e comunicação via rede de computadores
Cliente disponível	O principal pesquisador (ou coordenador do projeto) cumpre o papel de cliente e toma as decisões de projeto. Os requisitos são discutidos pelos membros dos grupos de pesquisa
Conhecimento coletivo	Os grupos de pesquisa mantêm uma base de conhecimento coletiva e fazem referências constantes ao repositório <i>online</i> (inclusão, alteração e exclusão de documentos)
Jogo do planejamento	A comunicação síncrona e assíncrona entre os membros que participam do projeto é fundamental para a execução desta prática. Pode ser precedido por um jogo de metáforas ou por um <i>brainstorming</i> , que ajudam a definir e priorizar o que precisa ser implementado (refinar e definir as estórias dos usuários)
Metáfora	São utilizadas para gerar estórias dos usuários. Evoluem enquanto o código está sendo escrito
Quarenta horas por semana	Além das atividades de desenvolvimento dos projetos, os pesquisadores possuem muitas outras tarefas, por exemplo, redigir documentos técnicos, solicitar financiamento à agências de fomento, etc. Como resultado, o número de horas de trabalho por semana pode variar consideravelmente
Padrões de código	O uso de padrões de código é fundamental para disciplinar o desenvolvimento de projetos por grupos. Pode ser considerado como um dos fatores que torna o software mais compreensível
Pontos de controle	Definindo-se pontos de controle, as divergências dos principais objetivos de pesquisa podem ser analisadas periodicamente e decisões são tomadas considerando-se o contexto geral dos projetos
Testes	Testes de unidade são executados, como acontece em XP, para assegurar a qualidade e a validade dos resultados de pesquisa
Refatoramento	Ocorre ao final de cada iteração de desenvolvimento
Modelagem baseada em componentes	Promove a disponibilização do conhecimento sobre o projeto aos grupos

como CVS², *Rational Rose*³ e o *framework* de testes JUnit⁴. Pretende-se incluir, ainda, ferramentas de rastreamento de bugs (*bug trackers*), ferramentas de auxílio à votação,

²Concurrent Versions System; <http://www.nongnu.org/cvs/>

³<http://www-306.ibm.com/software/rational/>

⁴<http://www.junit.org/index.htm>

dentre outras. Mais recentemente foi disponibilizado o EcliXPWeb⁵, que é um *plug-in* para o ambiente Eclipse⁶ e que faz referências às bases de dados do XPWeb.

Podem ser destacados como elementos favoráveis ao método XR:

1. o fato de considerar o desenvolvimento de projetos distribuídos, que está sendo observado como uma tendência em termos de projetos de software no geral e é um requisito em termos de processos para o desenvolvimento de pesquisas;
2. o desenvolvimento e a evolução de um portal que oferece suporte à execução das práticas e integra ferramentas que auxiliam a execução de diferentes atividades do processo;
3. está baseado em um método difundido (XP), conhecida pela comunidade científica.

Como elementos desfavoráveis, observa-se que:

1. não foram indicadas as atividades a serem executadas e os artefatos a serem gerados;
2. não foi indicado como são tratados elementos fundamentais de sistemas distribuídos, por exemplo, direitos de acesso e políticas de uso de um repositório de dados;
3. a proposta trata, fundamentalmente, de um modelo de ciclo de vida para desenvolvimento de projetos de pesquisa. Há poucas referências à atividades de garantia de qualidade, atividades de apoio organizacional.

2.8.2 Processo padrão para pesquisa e desenvolvimento

Foi proposto em 2006 um processo padrão para pesquisa e desenvolvimento por pesquisadores do Instituto de Pesquisas em Telecomunicações e Eletrônica na Korea (Hwang e Park, 2006). Padrões internacionais para Engenharia de Sistemas e Engenharia de Software e a própria experiência dos pesquisadores foram considerados.

Conforme apresentado na Figura 2.2, o processo é composto por 41 sub-processos, divididos em quatro categorias que são: (1) processos de ciclo de vida para projetos de pesquisa e desenvolvimento, (2) processos de suporte, (3) processos de gerenciamento de projetos e (4) processos organizacionais. A primeira categoria é dividida em cinco sub-categorias, que são: sistemas/software, dispositivos, tecnologias, padrões, políticas e estratégias. O objetivo desta categoria é auxiliar o desenvolvimento de produtos que atendam os requisitos dos clientes. A categoria de processos de suporte foi proposta para assegurar a integridade dos artefatos produzidos durante o desenvolvimento do projeto. A categoria de processos de gerenciamento de projetos estabelece os planos de projeto que precisam ser elaborados e avalia os resultados e os progressos obtidos em relação

⁵<http://sourceforge.net/projects/eclixpweb>

⁶www.eclipse.org

aos planos. A categoria de processos organizacionais estabelece os objetivos de negócios relacionados ao projeto e propõe atividades que ajudem a alcançar tais objetivos.

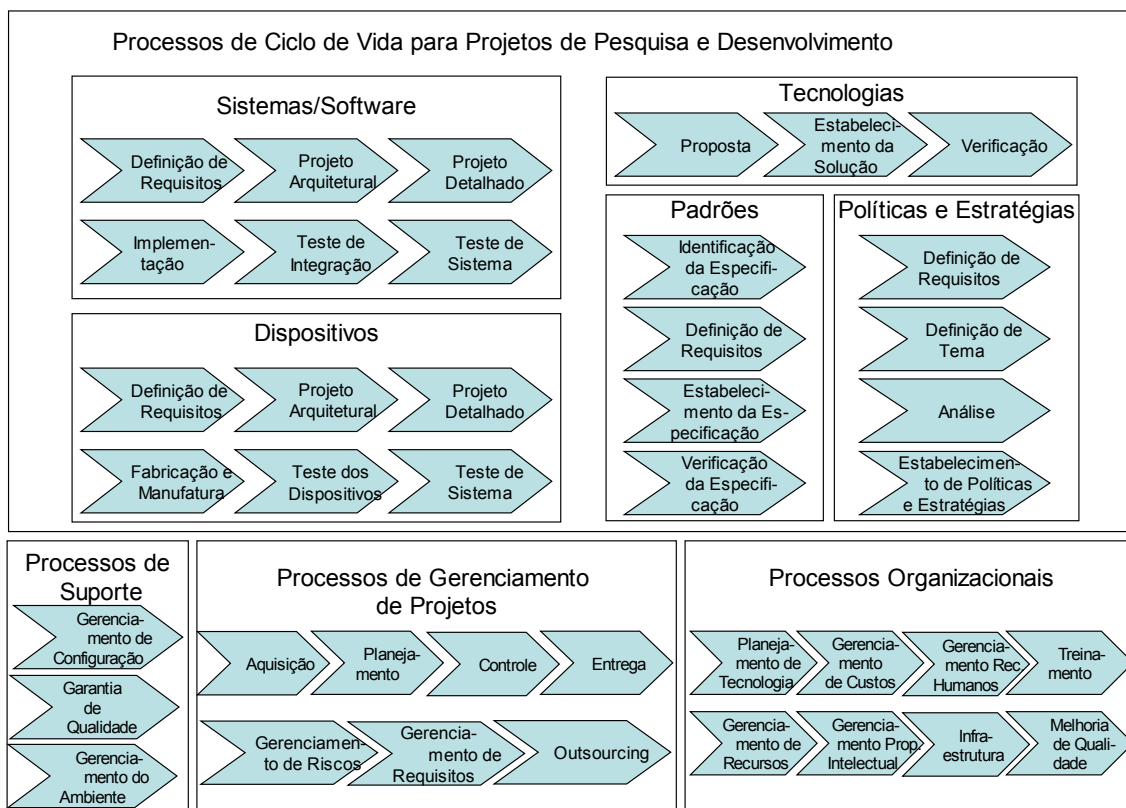


Figura 2.2: Visão geral do processo para pesquisa e desenvolvimento proposto por Hwang e Park (2006)

Observa-se, como um dos principais pontos favoráveis, que o processo proposto é amplo no sentido de englobar não somente atividades de ciclo de vida, mas também atividades de gerenciamento do projeto, de garantia de qualidade e de engenharia de sistemas. Além disso, foram considerados interesses do setor industrial para a elaboração da proposta, cobrindo um dos principais requisitos de um processo para o desenvolvimento de projetos de pesquisa, que está relacionado à participação do setor industrial no desenvolvimento de pesquisa (discutido na Seção 4.4).

Pode ser mencionado, como elemento desfavorável, o fato de a proposta estar direcionada ao contexto da organização em que os autores do proposta trabalham. Não foram considerados trabalhos de outros pesquisadores, nem requisitos apresentados na literatura sobre processos para pesquisa e desenvolvimento. Uma consequência disso é que processos considerados fundamentais por outros autores, por exemplo, aqueles relacionados à manutenção dos projetos e a colaboração entre os membros, não foram incluídos. Finalmente, não foram apresentadas informações sobre as possibilidades e as restrições da generalização da proposta para outros contextos diferentes daquele para o qual a proposta foi elaborada.

2.8.3 HDG (*Higher Degree Process*)

O HDG (Walker, 2003) foi proposto por pesquisadores do SEAL (*Software Engineering Applications Laboratory*), da África do Sul, com o objetivo de ser utilizado como um modelo de referência para a realização de avaliações dos processos executados no próprio laboratório. É interessante notar o interesse dos pesquisadores com a qualidade e a melhoria dos processos do laboratório. Após conseguir a certificação da norma ISO 9001 (1994), em 1995, tiveram a iniciativa de avaliar seus processos de acordo com as recomendações da norma ISO/IEC 15504 (e para isso propuseram o HDG). Em 2001, por exemplo, foi alcançado o nível 3 para alguns dos processos do laboratório. Poucos anos depois foi alcançado o nível 5 para outros processos. O HDG foi um dos resultados obtidos com os esforços de melhoria de processos de software no contexto do SEAL. As práticas base que compõem o processo são apresentadas na Tabela 2.2.

Tabela 2.2: Práticas base do HDG (Walker, 2003)

HDG.PB1 – Definir os objetivos do projeto de pesquisa: definir uma proposta de projeto de pesquisa, indicando os objetivos, os principais requisitos, os recursos necessários, os prazos, os requisitos técnicos, os requisitos de qualidade de processos e os produtos de trabalho que deverão ser gerados
HDG.PB2 – Planejar a pesquisa: identificar as entradas necessárias, as atividades a serem cumpridas, os recursos, os prazos associados às atividades, as responsabilidades dos membros e os relatórios necessários
HDG.PB3 – Determinar as condições técnicas: revisar a literatura técnica e avaliar o impacto do projeto
HDG.PB4 – Realizar revisão detalhada dos objetivos técnicos do projeto: revisar o impacto do resultado da revisão da literatura nos objetivos do projeto, atualizando o plano, se necessário
HDG.PB5 – Desenvolver o projeto de pesquisa: cumprir as atividades apresentadas no plano de projeto
HDG.PB6 – Revisar e aprovar os resultados: o coordenador do projeto e os pesquisadores revisam os produtos de trabalho resultantes do cumprimento das atividades propostas. Os produtos são aprovados quando os objetivos são alcançados e os critérios de qualidade são satisfeitos
HDG.PB7 – Validar os resultados da pesquisa em relação a proposta: assegurar que os produtos de trabalho resultantes do projeto de pesquisa cumprem os requisitos técnicos
HDG.PB8 – Disseminar os resultados da pesquisa: apresentar os resultados da pesquisa no formato de artigos técnicos, de forma que possam ser revisados, apresentados em conferências e reuniões técnicas e publicados em meios reconhecidos
HDG.PB9 – Redigir a dissertação ou a tese: produzir o documento de dissertação ou tese a partir dos produtos de trabalho do projeto
HDG.PB10 – Avaliar os resultados do projeto de pesquisa em relação aos requisitos de avaliação: a dissertação e a tese são avaliadas em relação aos requisitos iniciais do projeto

Como elemento favorável ao processo proposto, pode-se destacar o fato de o autor ter apresentado os processos indicando atividades e artefatos que são intrínsecos ao desenvolvimento de pesquisas. Outra parte do processo apresentado, que complementa as práticas base, se refere à descrição de um conjunto de problemas que geralmente ocorrem

no desenvolvimento de pesquisas (por exemplo, o pequeno número de publicações obtidas como resultado dos projetos de pesquisa) e oportunidades de melhoria. As ações sugeridas são bastante simples e, adotá-las na prática pode ser muito útil para dar início a programas de melhoria de processos em outras instituições ou grupos de pesquisa.

Como elemento desfavorável pode-se citar a descrição bastante genérica da proposta. Não foram apresentados detalhes sobre a execução das práticas base (devem ser executadas sequencialmente? de forma iterativa? são compostas por quais atividades?) e dos artefatos a serem elaborados.

2.8.4 A proposta de Nunamaker e Chen

Nunamaker e Chen (1990) propuseram um processo para desenvolvimento de sistemas de informação no contexto de projetos de pesquisa. Para a elaboração da proposta, os autores consideraram a importância em integrar elementos de metodologia de pesquisa em um *framework*, abrangendo o desenvolvimento de um sistema. Uma visão geral da proposta é apresentada na Figura 2.3.

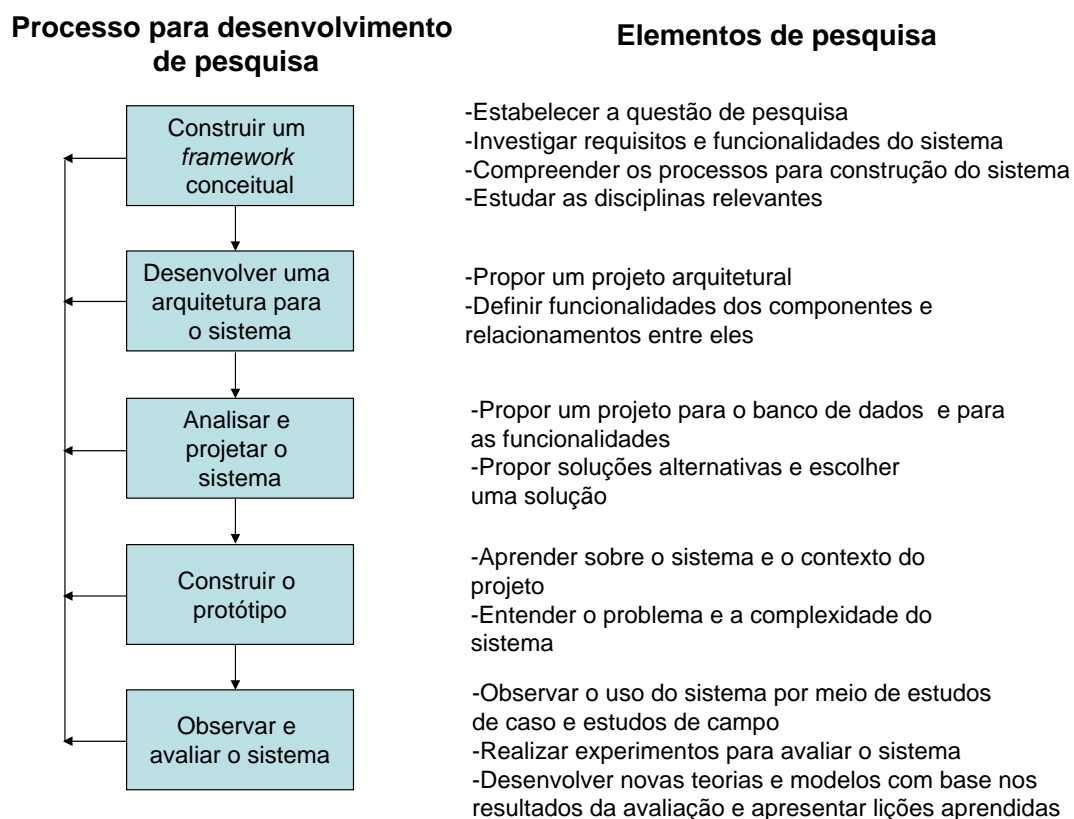


Figura 2.3: Processo para desenvolvimento de projetos de pesquisa proposto por Nunamaker e Chen (1990)

Para a fase de construção de um *framework* conceitual, os autores sugerem que os pesquisadores avaliem o quanto a questão de pesquisa é significativa. Além disso, devem

ser discutidos os métodos que serão usadas para resolver o problema. Na fase de elaboração da arquitetura, é importante apresentar o sistema sob uma perspectiva de componentes, especificar suas funcionalidades e definir a estrutura e as interações dinâmicas entre os componentes. Na fase de análise e projeto do sistema deve-se compreender o domínio do problema, observar a possibilidade de aplicação de conhecimento técnico e científico relevante e as alternativas para o projeto. Na fase de construção do protótipo, a implementação do sistema é realizada para demonstrar a viabilidade do projeto de pesquisa. Na fase de observação e avaliação do sistema, os pesquisadores avaliam o desempenho e a usabilidade do sistema conforme definido nas fases iniciais do desenvolvimento.

O principal *fator positivo* observado na proposta de Nunamaker e Chen é a integração de elementos de metodologia para desenvolvimento de projetos de pesquisa. Considerando-se que a proposta foi apresentada no começo da década de 1990, nota-se que foi uma iniciativa importante no sentido de divulgar elementos de metodologia científica no contexto de um processo para desenvolvimento de projetos de pesquisa. O principal *fator negativo* observado é a ausência de contextualização da proposta, ou seja, não foram mencionados os requisitos considerados na elaboração da proposta e como a avaliação foi realizada. Nota-se, ainda, que o processo cobre apenas atividades de desenvolvimento e não faz referência a atividades como gerenciamento de projetos e garantia de qualidade. Também não é mencionada a possibilidade de desenvolver projetos distribuídos e colaborativos, que são tendências em termos de desenvolvimento de pesquisas, usando o método proposto.

2.8.5 *Collaboratories*

De forma geral, *collaboratories* estão relacionados ao uso de tecnologias de colaboração para apoiar o desenvolvimento de pesquisas científicas geograficamente distribuídas. Formalmente, são definidos como “um centro sem paredes”, em que pesquisadores podem desenvolver suas pesquisas sem se preocupar com a localização geográfica (Wulf, 1989). Alguns pesquisadores registraram lições aprendidas como resultado de suas experiências com *collaboratories* e indicaram atividades e ferramentas que consideraram importantes, conforme apresentados a seguir:

- Olson et al. (2002) apresentaram três elementos que consideram fundamentais para que a colaboração (e a implantação de *collaboratories*) possa ocorrer: motivação das pessoas envolvidas para o trabalho em equipe, colaboração constante e disponibilidade de infra-estrutura e de ferramentas. Os autores destacaram como lições aprendidas a importância em (1) usar um método iterativa, em que os ciclos de desenvolvimento envolvam as fases de projeto, desenvolvimento, implantação e avaliação; (2) prover e utilizar ferramentas que promovam a participação dos membros

em seminários e reuniões de laboratórios virtuais (3) discutir normas éticas das comunidades ou países participantes e (4) oferecer treinamento sobre ferramentas e tecnologias antes que sejam adotadas para uso. Os autores experimentaram diferentes ferramentas em suas experiências em *collaboratories* e destacaram as seguintes: Microsoft NetMeeting⁷, Timbuktu Conference⁸ e Virtual PC⁹.

- Sonnenwald et al. (2003) realizaram experimentos em *collaboratories* e indicaram atividades que consideraram ser primordiais a partir do estudo que realizaram: fornecer canais redundantes para promover a comunicação, fornecer múltiplas formas para visualização de vídeo, compartilhar instrumentação científica, fornecer canais satisfatórios para transmissão de áudio, fornecer informações sobre o que os membros do projeto estão fazendo, oferecer dicas sobre o *status* dos membros em relação à comunicação (por exemplo, se estão *online* ou *offline*),
- O trabalho de Farooq et al. (2005) também deve ser mencionado, pois foi investigada a relação entre criatividade, que é um aspecto central do desenvolvimento de pesquisas, e o suporte à colaboração por meio de ferramentas. Como um dos resultados da pesquisa, foi proposto um protótipo para ser usado em ambiente de *collaboratories* denominado BRIDGE (*Basic Resources for Integrated Distributed Group Environments*¹⁰). O protótipo disponibiliza funcionalidades para troca de mensagens, suporte à geração de gráficos, suporte a *social awareness* e *workspace awareness* e coordenação e planejamento dos projetos. Além disso, está disponível uma linha de tempo (*timeline*) em que é indicado o histórico do projeto e os *deadlines* importantes. O sistema apresenta uma integração entre os gráficos, os documentos e a linha de tempo.

O objetivo dos *collaboratories* é permitir que cientistas que estão geograficamente dispersos possam trabalhar de forma colaborativa, usando tecnologias apropriadas para acessar ferramentas, bancos de dados e instrumentos uns dos outros (Kouzes et al., 1996; Wulf, 1989). Assim, esse objetivo é um fator *positivo* da iniciativa.

No entanto, como fator *negativo*, não foi encontrado na literatura um processo, envolvendo atividades, infra-estrutura e gerenciamento de recursos, que tenha sido proposto para *collaboratories* e que tenha sido avaliado.

⁷<http://www.microsoft.com/windows/netmeeting/>

⁸(<http://www.netopia.com/software/products/tb2/>)

⁹(<http://www.apple.com/macosex/applications/virtualpc/>)

¹⁰<http://bridgetools.sourceforge.net>

2.9 Considerações finais

Neste capítulo buscou-se apresentar os principais métodos em termos de processos de software e modelos de qualidade que foram importantes como fundamentação científica para o desenvolvimento deste trabalho. Os processos para desenvolvimento de software são relevantes neste contexto, pois considera-se que um dos artefatos essenciais que permitem a realização de provas de conceito em termos de pesquisas é o software, o protótipo que tenha sido construído no projeto de pesquisa.

Além disso, os processos para desenvolvimento de projetos distribuídos e colaborativos, os processos de software livre e os métodos ágeis também fornecem elementos que foram importantes para o desenvolvimento deste trabalho e, portanto, foram descritos. Por fim, foram apresentados os trabalhos encontrados na literatura que dão ênfase em processos para desenvolvimento de projetos de pesquisa.

Design Rationale

3.1 Considerações iniciais

Neste capítulo são apresentados conceitos sobre *design rationale*, considerando-se suas principais perspectivas e as atividades de captura, representação e recuperação das informações. Foram enfatizados também os trabalhos de pesquisa apresentados na literatura, com o objetivo de apresentar os principais resultados que já foram obtidos e as principais tendências em relação ao desenvolvimento de novas pesquisas. É importante destacar que os conceitos de *design rationale* estão sendo aplicados em diferentes áreas de conhecimento e, neste capítulo, o foco apresentado foi a disciplina de Engenharia de Software.

3.2 Visão geral sobre *design rationale*

De acordo com Gruber e Russel (1991) e Moran e Carroll (1996), *design rationale* se refere a: (1) o **raciocínio** que justifica um projeto resultante e (2) as descrições que justificam porque determinadas estruturas foram escolhidas sobre as demais **alternativas**. Lee (1997) considera que *design rationale* abrange não apenas as decisões de um projeto, mas também as justificativas, as alternativas e as argumentações que levaram à decisão. Essas informações adicionais valorizam a documentação do processo e colaboram para o aprendizado do projeto como um todo.

Historicamente, os estudos envolvendo *design rationale* começaram a ser realizados a partir da divulgação do trabalho de Rittel, em 1970, sobre um modelo para auxiliar o registro de argumentações de projeto, denominado IBIS (*Issue-Based Information System*) (Kunz e Rittel, 1970) apud (Regli et al., 2000). O modelo foi aplicado durante

duas décadas em projetos de larga escala e em projetos de engenharia e arquitetura. Em meados da década de 1980, Conklin e Begeman (1988) propuseram um sistema de hipertexto, denominado gIBIS (*graphical IBIS*), para facilitar a captura colaborativa de *design rationale*. Outros modelos relacionados à modelagem da argumentação foram propostos em seguida como variações do IBIS, por exemplo, o modelo PHI (*Procedural Hierarchy of Issues*) (McCall, 1991) e o modelo QOC (*Questions, Options and Criteria*) (MacLean et al., 1991) (estes modelos serão descritos na Seção 3.3.2).

A importância de *design rationale* em Engenharia de Software tem sido reconhecida desde a década de 1980 (Parnas e Clements, 1986; Potts e Bruns, 1988). A partir dessa época, muitas pesquisas começaram a ser realizadas com o objetivo de aplicar a abordagem em diferentes fases do processo de desenvolvimento de software (Bass et al., 2006; Canfora et al., 2000; Myers et al., 1999; Sutcliffe e Ryan, 1998). Dutoit et al. (2006) observaram que os conceitos de *design rationale* podem ser aplicados nas diversas atividades de Engenharia de Software, e apresentaram alternativas, ainda bastante genéricas, para utilizá-los nos contextos das atividades apresentadas na norma ISO/IEC 15504: aquisição e fornecimento, engenharia, operação, suporte, gerenciamento, reuso, melhoria de processo e recursos e infra-estrutura. Não há, atualmente, trabalhos que cubram a aplicação de *design rationale* em todas as atividades mencionadas. É interessante notar, em especial, a concentração de pesquisas sobre a aplicação de *design rationale* na fase de engenharia de requisitos. Em alguns estudos de caso realizados (Conklin e Burgess-Yakemovic, 1996; Karsenty, 1996; Paiva e Fortes, 2005), foi observada sua importância como um mecanismo para auxiliar a estruturação de informações que terão impacto em todo o projeto. Dutoit e Paech (2000) relacionaram a importância de *design rationale* ao fato de que erros ou mudanças nos requisitos afetam todo o desenvolvimento do projeto e os custos e, por isso, precisam ser amplamente discutidos e documentados.

Desde o surgimento dos primeiros estudos sobre *design rationale*, diversos benefícios foram apresentados por pesquisadores em relação à adoção da abordagem na prática. De acordo com Burge e Brown (2000), *design rationale* é importante para auxiliar a execução das atividades de revisão, manutenção, documentação, avaliação e aprendizado de um projeto. Além disso, pode ser bastante útil para reuso de projetos passados (Ball et al., 1999; Stutt e Motta, 1995), para coordenação de pessoas que fazem parte de um grupo de trabalho (Conklin e Begeman, 1988), para promoção de reflexões críticas durante o desenvolvimento de projetos (Fischer et al., 1989), para manutenção de artefatos (Boy, 1995), para auxiliar o gerenciamento dos projetos e a colaboração entre os membros, melhorar a documentação e auxiliar na fase de engenharia de requisitos (Dutoit et al., 2006).

Por outro lado, Shipman e McCall (1997), Regli et al. (2000) e Conklin e Burgess-Yakemovic (1996) apresentaram limitações das abordagens de *design rationale* que dificultam sua adoção na prática: dificuldades encontradas pelos desenvolvedores na

recuperação das informações capturadas, diferenças entre o tipo de informação que o desenvolvedor percebe a necessidade (ou gostaria) de registrar e o que o modelo permite; a generalidade das ferramentas, por não suprirem as necessidades particulares de um ambiente ou de uma empresa e a interferência à progressão natural das atividades de projeto.

Os sistemas de *design rationale* têm como objetivo enriquecer o projeto auxiliado por computador (*Computer Aided Design* - CAD) e outras atividades da engenharia. Para melhor aproveitamento, é preciso integrá-los a outros sistemas de suporte ao projeto, por exemplo, sistemas de suporte à tomada de decisão e ferramentas CASE (*Computer-Aided Software Engineering*). De modo geral, as funcionalidades de um sistema de *design rationale* são a captura, a representação e a recuperação de *design rationale*. A captura refere-se à obtenção de informação sobre as decisões de projeto e a conversão em um formato lógico. A representação faz uso de um esquema para representar o *design rationale* que será armazenado. E a recuperação é responsável pelo fornecimento das informações adequadas de *design rationale* de acordo com as solicitações do usuário (Regli et al., 2000).

3.3 Atividades envolvendo *design rationale*

Muitos trabalhos relacionados à captura, representação e recuperação de *design rationale* começaram a ser desenvolvidos a partir de 1980. Alguns deles são apresentados nas sub-seções seguintes.

3.3.1 Captura de *design rationale*

A captura de *design rationale* geralmente ocorre a partir da comunicação entre membros de uma equipe de projeto que, na maioria das vezes, utilizam ferramentas de trabalho cooperativo suportado por computador. Essa captura pode ocorrer de forma automática, utilizando-se ferramentas de *design rationale* especializadas, ou de forma interativa, em que um membro da equipe de projeto fica responsável por registrar e estruturar as informações consideradas relevantes. As informações capturadas são organizadas por meio de regras (determinadas por um esquema de representação) e são armazenadas em um repositório de projeto com o objetivo de permitir a recuperação das mesmas quando necessário.

Durante o desenvolvimento de um projeto, as informações de *design rationale* são capturadas registrando-se o raciocínio, as decisões e as alternativas de projeto com o objetivo de construir uma estrutura formal ou semi-formal e viabilizar a recuperação posterior das informações. O processo de captura de *design rationale* geralmente consiste de duas etapas: *registro do conhecimento*, que envolve a captura da maior quantidade de

informação possível, e *construção do conhecimento*, que se refere à extração, organização e armazenamento do conhecimento (Regli et al., 2000).

Um problema que diversos pesquisadores têm tentado resolver, neste contexto, é a determinação de qual informação capturar durante o projeto e como capturá-la: capturar pouca informação (ou a informação errada) dificulta a representação das informações de *design rationale* e traz poucos benefícios aos projetistas e capturar informações de forma intrusiva não motiva os projetistas a usar o sistema de *design rationale*. Um requisito básico é que sejam capturadas informações em um formato que permita a comunicação e o reuso de conhecimento de projeto (Regli et al., 2000).

Do ponto de vista do usuário, os métodos de captura de *design rationale* podem ser divididos em duas categorias: aquela que requer intervenção do usuário (interativa), em que é preciso registrar manualmente as informações de projeto durante o processo de desenvolvimento do sistema; e aquela que é automática, em que a captura é executada por uma ferramenta.

No método interativo, registram-se, basicamente, quais decisões foram tomadas, quando e porque foram tomadas e quem foram os responsáveis por elas (Shipman e McCall, 1997). A vantagem desta forma de captura é que a informação armazenada (semi-formal ou formalmente) é estruturada, o que facilita a atividade de recuperação. A desvantagem refere-se ao fato de que os usuários são desviados de suas atividades usuais.

No método de captura automática, assume-se a utilização de uma ferramenta automatizada para capturar a comunicação que ocorre entre membros de um grupo de trabalho. Os registros obtidos podem então ser utilizados para extrair as informações de *design rationale* resultantes da atividade de projeto. A vantagem dessa forma de captura é que os desenvolvedores não precisam se envolver diretamente na tarefa de captura da informação. A desvantagem refere-se ao fato de a informação armazenada não ser estruturada o que gera, muitas vezes, insucesso na recuperação quando uma consulta é realizada.

Algumas propostas foram apresentadas como resultado de projetos de pesquisa para a captura de *design rationale*. Gupta et al. (2001) propuseram o modelo DRIMER (*Design Recommendation and Intend Model Extended for Reusability*) com o objetivo de auxiliar as atividades de captura, representação e reuso de *design rationale*. Ao utilizar o modelo, uma equipe de projetistas apresenta propostas de projeto considerando os objetivos, as restrições, as funções e as metas do mesmo. Podem ser apresentadas diferentes propostas e cada uma delas pode consistir de sub-propostas, de forma a reduzir a complexidade. Além disso, uma proposta pode suportar, contradizer ou mudar as idéias defendidas em outra proposta já existente, representando a natureza argumentativa de projetos colaborativos. Em cada proposta, devem ser incluídas as recomendações que justificam a implementação daquela proposta em particular, representando um relacionamento entre um problema e uma solução. Gupta et al. acreditam que o modelo proposto oferece um

importante mecanismo para o desenvolvimento de software por integrar a utilização de *design rationale* ao processo do projeto e ajudar o projetista a tomar decisões. A captura de *design rationale* ocorre como uma atividade natural do processo de software em um meio colaborativo.

Kato et al. (2002) propuseram um sistema denominado IDIMS (*Integrated Design Information Management System*) desenvolvido para capturar o *design rationale* resultante da comunicação por *email* entre membros de uma equipe de projeto. O sistema é formado pelos seguintes componentes: repositório de documentos, repositório de questões e decisões e gerenciador de *emails*. O repositório de documentos armazena todos os documentos produzidos durante o processo de desenvolvimento de um projeto. Os usuários podem adicionar, procurar e recuperar documentos por meio de uma interface *web*. O repositório de questões e decisões armazena anotações sobre questões e decisões relacionadas aos documentos. Os usuários podem adicionar, procurar e visualizar as questões e decisões armazenadas utilizando uma aplicação *windows*, por meio da qual a estrutura da argumentação é apresentada ao usuário. O gerenciador de *emails*, implementado como um filtro de *emails*, faz o processamento de mensagens que são trocadas entre membros de uma equipe. Ele é configurado para receber *emails* enviados para uma determinada lista. Ao receber um *email*, o filtro transforma a mensagem em um documento XML e armazena-o em uma base de dados. As anotações do documento (questões) são extraídas e armazenadas no repositório de questões e decisões, possuindo um identificador. O gerenciador então envia a mensagem aos membros da lista e anexa um *template* para resposta, que cita a mensagem original e as anotações. Essas anotações possuem referências para os identificadores previamente estabelecidos. Como resultado, são criados *links* entre questões e decisões discutidas por *email* de uma forma semi-automática.

3.3.2 Representação de *design rationale*

Para a representação de *design rationale* previamente capturado, é necessário que seja concebida uma estruturação das informações capturadas durante o desenvolvimento dos projetos, em geral, provenientes da discussão entre os membros da equipe. De acordo com Shipman e McCall (1997), *design rationale* pode ser observado sob diferentes perspectivas, denominadas argumentação, comunicação e documentação, sendo que a forma de representação das informações varia de acordo com a perspectiva adotada pela equipe de trabalho.

Na perspectiva da **argumentação**, considera-se que o *design rationale* está relacionado ao raciocínio de projetistas para a resolução de problemas e a estruturação de soluções, independentemente se eles trabalham individualmente ou em grupo. Nesta perspectiva

são documentados também os argumentos apresentados pelos participantes durante a discussão do projeto.

De acordo com a perspectiva da **comunicação**, *design rationale* está relacionado à captura e à recuperação de toda comunicação que ocorre entre os desenvolvedores. Por exemplo, arquivos de áudio e vídeo, *emails*, anotações e diagramas são armazenados. Nesta perspectiva, o registro da comunicação entre os projetistas não deve causar qualquer impacto no processo de desenvolvimento do projeto. O objetivo principal é registrar a troca de mensagens entre os participantes de um projeto.

Na perspectiva da **documentação**, *design rationale* se refere ao registro da informações sobre as decisões de um projeto, ou seja, quais decisões foram tomadas, quando foram tomadas, quem foi o responsável e quais foram os motivos que levaram àquela decisão. Em geral, apenas as decisões finais e uma breve explicação das mesmas são registradas. Portanto, a perspectiva da documentação sugere o registro de menos informações em relação às perspectivas da argumentação e comunicação, por considerar que é fundamental registrar apenas as decisões tomadas e não o raciocínio utilizado para o desenvolvimento do projeto.

As perspectivas mencionadas utilizam diferentes formas de representação de *design rationale*. A perspectiva de comunicação pode utilizar qualquer tipo de mídia para esse fim e a perspectiva de documentação faz uso de um esquema linear de representação. A perspectiva de argumentação utiliza nós e *links* para a representação de *design rationale* e um modelo de argumentação, tais como o IBIS (Kunz e Rittel, 1970) apud (Regli et al., 2000) ou o PHI (McCall, 1991) são utilizados.

O modelo IBIS faz uso de três abstrações para representar o conhecimento, denominadas questão, posição e argumento. Tais abstrações são relacionadas entre si para representar a discussão dos desenvolvedores. Assim, as dúvidas que surgem durante o projeto são tratadas como questões, seguidas por uma ou mais posições que as respondem. Os argumentos podem suportar ou rejeitar as posições, como apresentado na Figura 3.1. Os relacionamentos também são pré-determinados pelo modelo.

O modelo IBIS é considerado pelos usuários como um modelo muito importante. Projetistas que trabalham sozinhos afirmam que a organização de idéias em termos de questão, posição e argumento ajuda a prestar mais atenção nas partes mais difíceis e críticas do problema, além de ajudar a perceber mais rapidamente as idéias que são incoerentes. Aqueles que trabalham em equipe consideram que a estrutura imposta é muito útil e serve para mostrar opiniões dos participantes de forma clara (Monk et al., 1995). Como restrição do IBIS observa-se que, por ser um modelo simples, diversas informações ficam armazenadas em um único nó. Por exemplo, pode ser gerada uma rede complexa quando muitos nós são armazenados. Muitas vezes não se sabe qual questão foi armazenada primeiro e a qual questão determinado argumento pertencia inicialmente (Wiegeraad, 1999).

que permitem estruturar o espaço de alternativas, as opções fornecem possíveis respostas às questões e os critérios são a base para avaliar e escolher uma das opções.

Enquanto os sistemas que usam os modelos IBIS e PHI têm o propósito de capturar o histórico das deliberações, os sistemas que usam o modelo QOC enfatizam o desenvolvimento sistemático de um espaço de opções de projeto estruturado por questões (Regli et al., 2000).

Conforme apresentado na Figura 3.3, os possíveis relacionamentos entre os nós do QOC são: uma opção *responde a* uma questão; uma questão *é consequência de* uma opção; um critério faz uma *avaliação positiva* de uma opção e um critério faz uma *avaliação negativa* de uma opção. Assim que um nó *questão* é inserido, vários nós *opção* são inseridos de forma a tentar solucionar a questão. Para avaliar se uma *opção* é válida, são usados os nós *critério*, com o objetivo de avaliar uma *opção* ou mais. Sendo assim, tal avaliação pode ser positiva ou negativa para a *opção*. Uma mesma *opção* pode ter um determinado critério a seu favor e outro contra.

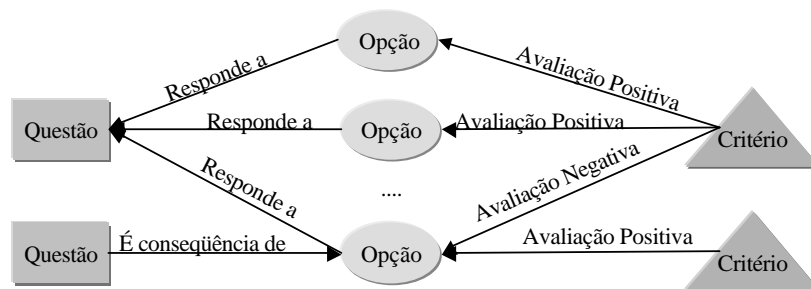


Figura 3.3: Abstrações e relacionamentos do modelo QOC (MacLean et al., 1991)

Conforme apresentado por Lee (1997), a representação de *design rationale* pode ser informal, semi-formal ou formal. A escolha de uma determinada abordagem depende dos serviços que o sistema deverá fornecer. Na representação informal, *design rationale* é capturado de uma forma não estruturada: são capturadas descrições em linguagem natural, registros de áudio e vídeo e rascunhos de diagramas. Geralmente são fáceis de criar, mas os sistemas computacionais não podem interpretá-los.

Em uma representação semi-formal, apenas partes do conteúdo pode ser interpretada computacionalmente. No entanto, os sistemas que implementam este tipo de representação podem suportar muitos serviços computacionais cujas implementações não são possíveis de implementar quando a representação informal é utilizada. Além disso, há menos sobrecarga aos projetistas na atividade de captura de *design rationale* porque o sistema pode sugerir quais informações são esperadas.

Em uma representação formal, objetos e relações são definidos como objetos formais que o sistema pode interpretar e manipular usando operações formais. A captura de *design rationale* torna-se uma forma de criar uma base de conhecimentos utilizando-se

uma linguagem formal. Observa-se, ainda, que quanto mais formalmente representadas estiverem as informações de *design rationale*, mais serviços o sistema poderá fornecer automaticamente. O autor sugere a utilização de uma abordagem incremental como forma de minimizar o custo da formalização do conhecimento, ou seja, transformando uma representação semi-formal em uma representação formal.

Conforme observado por Wiegeraad (1999), os modelos apresentados compartilham as mesmas estruturas básicas de “problema de projeto”, “solução alternativa”, “argumentos” e “critérios”, ou seja, diferentes termos são usados para representar basicamente os mesmos conceitos. Nota-se, também, que os modelos de representação são bastante genéricos, o que sugere a possibilidade de uma extensão ou adaptação dos mesmos de acordo com as peculiaridades do domínio de aplicação.

Ressalta-se o fato de que as pesquisas tendem a usar a perspectiva de argumentação e o modelo IBIS para a representação de *design rationale*. Possivelmente isto ocorre devido ao fato de ser o modelo mais genérico e que fornece uma estrutura básica que pode ser adaptada para outros ambientes.

3.3.3 Recuperação de *design rationale*

A recuperação de *design rationale* é determinada, principalmente, pelo esquema de representação utilizado. Algumas situações (cenários) que motivam a recuperação de *design rationale* se referem à visualização de projetos similares e suas decisões associadas; à recuperação de critérios, regras e opções que ajudem a tomar decisões durante o projeto e à produção de documentos após o processo de projeto (Regli et al., 2000).

Diferentes estratégias de recuperação podem ser utilizadas, dependendo dos interesses dos usuários do sistema. Regli et al. (2000) consideram três possíveis abordagens: navegação, pesquisa ou recuperação automática. Na abordagem da *navegação*, os projetistas podem verificar o *design rationale*, percorrendo nós por meio de *links* existentes entre eles. Para o projeto de um artefato complexo, geralmente uma grande quantidade de informação é armazenada, o que dificulta a busca por respostas específicas utilizando a estratégia de navegação. A recuperação por *pesquisa* é considerada mais eficiente que a recuperação com base em navegação. As consultas podem seguir a estrutura “o que acontece se” (*what-if*), que podem ser respondidas explorando-se diferentes opções; ou a estrutura “por que”, que são respondidas fazendo-se *backtracking* na rede de nós e *links* com a finalidade de encontrar a argumentação ou o raciocínio relacionado à decisão. Na recuperação *automática*, é feito um monitoramento do desenvolvimento de um projeto, de acordo com as regras, os critérios e as restrições definidas. Um monitor é utilizado para observar e verificar o processo de projeto e comparar as decisões tomadas com regras, critérios e restrições da biblioteca de *design rationale* ou da base de conhecimento. Quando são encontradas diferenças, o *design rationale* é apresentado automaticamente.

Em relação à recuperação de *design rationale*, os sistemas são classificados por Lee (1997) como sistemas de iniciativa do usuário ou de iniciativa do próprio sistema. No primeiro caso, o usuário decide quais partes do *design rationale* serão examinadas, quando e como serão visualizadas. Estes sistemas devem auxiliar os usuários a tomar conhecimento das informações armazenadas e tornar fácil o acesso às partes desejadas. No segundo caso, o sistema utiliza uma base de conhecimentos para decidir quando e como apresentar um *design rationale* (ou partes dele). Tais sistemas devem possuir uma base de conhecimentos que possa auxiliar efetivamente o sistema a tomar decisões e devem ser capazes de apresentar as informações de *design rationale* de forma não intrusiva.

Chung e Goodwin (1998) propuseram o sistema IDIS (*Integrated Design Information System*), que apresenta o projeto de um determinado sistema e as anotações relacionadas às entidades do projeto. Quando um projetista clica em um item de um diagrama, a especificação desse item é apresentada bem como todas as questões e comentários relacionados a ele. O sistema permite também que o projetista possa acompanhar o progresso do projeto e revisar todas as etapas cumpridas. Os usuários podem tratar o sistema como uma base de texto, de forma que possa ser realizada uma busca por palavras-chave para determinar o que foi discutido sobre um determinado tópico. Além disso, para responder questões relacionadas a “como atingimos determinado estágio”, o sistema apresenta todas as informações relevantes que influenciaram a escolha de uma alternativa particular de projeto. Para responder questões relacionadas a diferenças entre duas alternativas de projeto, o sistema fornece um resumo das discussões que levaram a diferentes alternativas de projeto, dados que fazem parte de um projeto e não fazem parte de outro e valores diferentes de parâmetros associados aos mesmos itens em diferentes projetos.

Burge e Brown (2002) observaram que grande parte das pesquisas atuais focaliza a captura e a representação de *design rationale*. Os autores acreditam que estas atividades são muito importantes, porém o sucesso das mesmas está relacionado à utilidade de *design rationale* para projetos em desenvolvimento, ou seja, o valor real de um *design rationale* só pode ser avaliado quando ele de fato for útil em um projeto. Capturar grande quantidade de informações detalhadas não é relevante se tais informações não forem acessadas. Por outro lado, se o *design rationale* é útil no desenvolvimento dos projetos da organização, há um maior incentivo para que o projetista se empenhe na captura das informações necessárias, particularmente se elas puderem ser utilizadas imediatamente. Além disso, conhecendo-se como as informações serão utilizadas, é possível determinar o que deve ser capturado e como o conteúdo obtido deve ser representado.

Na seção seguinte é enfatizado, portanto, o uso de *design rationale*, observando-se como os principais conceitos estão sendo aplicados na prática e os benefícios que a abordagem pode trazer para o ambiente de desenvolvimento de projetos de pesquisa.

3.4 Uso de *design rationale*

De acordo com Dutoit et al. (2006), *design rationale* pode ser utilizado principalmente com os seguintes propósitos:

- promover a colaboração entre membros de uma equipe;
- facilitar a manutenção e o reuso;
- melhorar a qualidade dos artefatos;
- transferir conhecimentos.

A seguir, são apresentados trabalhos de pesquisa que estão sendo realizados no sentido de explorar as potencialidades de *design rationale* em relação aos seus propósitos.

3.4.1 Visando colaboração

Membros de uma equipe podem usar um repositório de *design rationale* para compreender as atividades que os outros membros estão executando e como estas atividades podem impactar em seus próprios trabalhos. Outra possibilidade de uso é a exposição de diferentes pontos de vista e o raciocínio associado. Além disso, *design rationale* é importante para promover o projeto participativo e colaborativo, de forma que os usuários possam compreender o raciocínio dos projetistas, as questões que estão sendo tratadas pelos desenvolvedores e as alternativas que estão sendo consideradas (Dutoit et al., 2006).

Falessi et al. (2006) propuseram uma técnica para documentar razões das decisões de projeto (*design decision rationale*) e avaliaram experimentalmente seu impacto em relação à eficácia e eficiência quando membros de diferentes equipes que desenvolvem um projeto precisam colaborar para que uma decisão seja tomada quando há mudanças de requisitos. Os autores propuseram uma técnica de documentação denominada DGA (*Decision Goals and Alternatives*), que considera que as decisões de projeto dependem dos objetivos da decisão e de relacionamentos entre decisões. De forma geral, eles sugeriram que as seguintes informações sejam capturadas: requisitos funcionais, requisitos não funcionais (atributos de qualidade e restrições), objetivos de negócios e decisão.

Na abordagem proposta, o conceito *decisão* é refinado em dois sub-conceitos: tipo de decisão (DT) e alternativas de decisão (DA). DT se refere ao problema que a decisão deve solucionar e DA representa as alternativas disponíveis. Os autores realizaram experimentos com a abordagem proposta para avaliar os efeitos da abordagem em situações em que indivíduos e equipes precisavam tomar decisões. Para ambos os casos, os resultados obtidos indicaram que a eficácia melhorou quando a documentação segundo o formato DGA está disponível e a eficiência permaneceu inalterada.

3.4.2 Visando manutenção e reuso

Em geral, pesquisadores entendem que um dos principais usos para *design rationale* está associado à execução da atividade de manutenção. Pessoas responsáveis por fazer alterações precisam compreender os efeitos das mudanças, bem como as razões pelas quais os artefatos foram elaborados daquela determinada forma, e o conhecimento das razões de projeto ajudam nessa compreensão. De forma semelhante, antes que o software possa ser reusado é necessário que seja entendido e/ou modificado. Isso requer que o raciocínio dos desenvolvedores do projeto seja entendido e que os componentes, elementos ou módulos que podem ser reusados sejam identificados (Dutoit et al., 2006). Em um estudo realizado recentemente, Babar et al. (2006) observaram que, na prática, projetistas e desenvolvedores do setor industrial percebem a utilidade de *design rationale* e percebem também que os esforços dispendidos na captura das informações, tendo em vista os benefícios que podem ser obtidos na fase de manutenção, valem a pena.

Lougher e Rodden (1993) propuseram um sistema de documentação cujo objetivo é fornecer apoio a grupos de pessoas que executam atividades de manutenção de software. Para o desenvolvimento do sistema foi explorada a tecnologia de hipertextos juntamente com um mecanismo de anotação flexível, de forma que informações de manutenção sejam registradas em uma “memória” do projeto. As informações de *design rationale* relativas à manutenção são associadas à parte específica do sistema que está sendo mantida (por exemplo, adicionando comentários ao código fonte), podendo ser facilmente visualizadas e alteradas por outras pessoas envolvidas com o projeto. Os autores não fizeram restrições em relação à representação da informação. Como aspecto positivo desse trabalho, pode-se destacar a facilidade de uso: os mantenedores podem incluir as informações em um formato livre, e não precisam se preocupar em apresentá-la de acordo com as categorias disponíveis. Por outro lado, a abordagem utilizada pelos autores dificulta a recuperação de informações específicas.

Burge e Brown (2006) têm investigado como *design rationale* pode ser útil para auxiliar a manutenção de software. Foi proposto o modelo RATSpeak, baseado no modelo de argumentação DRL, para a representação do *design rationale*. Além disso, foi desenvolvida a ferramenta SEURAT (*Software Engineering Using Rationale*) com o objetivo de oferecer suporte automatizado ao uso de *design rationale* na fase de manutenção, utilizando o modelo RATSpeak. A ferramenta apresenta as informações de *design rationale* ao mantenedor e faz inferências para detectar problemas e inconsistências que possam indicar problemas no projeto. A ferramenta foi proposta como um *plug-in* do sistema Eclipse. Buscou-se, com isso, integrar o código com a informação de *design rationale* e assegurar que os mantenedores estarão cientes de que as informações estão disponíveis e podem ser acessadas facilmente.

3.4.3 Visando a melhoria de qualidade

O registro de *design rationale* auxilia a identificação das decisões tomadas pelas equipes que desenvolvem o projeto e na certificação de que as decisões são mutuamente consistentes. A rastreabilidade de requisitos também pode ser melhorada, pois é possível fazer ligações entre requisitos e partes de outros artefatos que estão relacionados a eles. Além disso, *design rationale* pode ser utilizado para auxiliar a execução de atividades como depuração, correção de problemas e extensão de funcionalidades de um artefato (Dutoit et al., 2006).

Balabko et al. (2005) investigaram a possibilidade de utilizar *design rationale* para auxiliar a reengenharia, o reuso e a melhoria de processos de negócios. Eles propuseram a captura de informações de *design rationale* para cada um dos papéis envolvidos na modelagem de processos de negócios. De acordo com os autores, em geral, processos de negócios são difíceis de compreender se não houver uma exposição clara do *design rationale* que justifica a estrutura do processo. Dessa forma, foi proposto um método que sugere aos projetistas de processos de negócios que definam explicitamente como as atividades foram agrupadas e como estão relacionadas na especificação do processo. As fases que compõem o método proposto para a modelagem de processos de negócios são:

1. definir as atividades que irão compor o processo, em que se propõe que informações de *design rationale* sejam capturadas utilizando-se modelos tradicionais;
2. definir atividades alternativas (para cada alternativa são especificadas restrições, que são consideradas partes do *design rationale*) e a relação de dependência entre todas as atividades que compõem o modelo;
3. escolher a alternativa adequada, na qual um projeto será baseado.

Os autores enfatizam que o método proposto contribui para a captura de *design rationale* devido a possibilidade de raciocinar explicitamente em termos de composição de atividades de um processo. Adotando o método, é possível explicitar quem é responsável por qual atividade e como elas são integradas no modelo de processos de negócios resultante. Um ponto positivo do trabalho é o fato de tratar *design rationale* sob a perspectiva de processos, diferentemente da maioria dos trabalhos, que enfatiza o produto. Outro ponto a ser destacado é o respaldo à execução do modelo de processo de negócios de forma colaborativa. Os autores sugeriram que, ao executar o processo, informações relacionadas às dimensões *o que*, *quem*, *onde* e *quando* sejam registradas. Uma restrição ao trabalho observada foi a falta de estudos experimentais envolvendo a proposta.

3.4.4 Visando a transferência de conhecimento

Para aprender a partir de experiências passadas é necessário compreender o caminho seguido para tomar decisões e os benefícios e os problemas associados a ele. *Design rationale* pode ser útil para auxiliar a reutilização de conhecimento, como alternativa para o registro de idéias e projetos que foram bem sucedidos ou que não geraram bom resultado. Outro uso importante para *design rationale* é no treinamento de pessoas que passam a compor uma equipe. Um novo membro pode compreender as razões para o estado corrente de um artefato antes de sugerir mudanças (Dutoit et al., 2006).

Conforme apresentado por Hagge et al. (2006), *patterns* estão relacionados a *design rationale* porque buscam capturar o conhecimento que possa ser reutilizado. Os autores apresentaram um mapeamento entre as categorias usadas para descrever um *pattern* e os nós do modelo QOC. Eles observaram, como principal diferença, o fato de que no modelo QOC, todo o conjunto de opções pode estar contido em um único esquema e, em relação a *patterns*, somente uma opção é proposta. O conjunto de soluções é apresentado em uma família de *patterns* tratando o mesmo problema.

3.5 *Design rationale* no desenvolvimento de projetos de pesquisa

Uma abordagem para registro de *design rationale* no contexto do desenvolvimento de pesquisas foi proposta por Kim et al. (1993). Esses autores propuseram uma adaptação do modelo IBIS para o desenvolvimento de pesquisa científica colaborativa. As justificativas para escolha do modelo IBIS foram de que ele:

1. foi modificado para ser usado em diversos domínios, principalmente naqueles em que diferentes pontos de vista sobre o mesmo problema precisam ser coletados,
2. foi projetado para capturar as razões das discussões dos grupos,
3. também pode ser utilizado para representar trabalho científico individual.

Conforme apresentado na Figura 3.4, os seguintes nós foram definidos na proposta de Kim et al. (1993): Q - *questão*, H - *hipótese*, AL - *alegação*, SA - *sub-alegação* e AR - *argumento*. No exemplo da parte inferior da figura, é representada uma questão Q, que possui duas hipóteses: H1 e H2. A hipótese H1 possui três alegações AL1, AL2 e AL3. A alegação AL1 possui duas sub-alegações SA1 e SA2. AR1, AR2, AR3 e AR4 são argumentos contra ou a favor as alegações e sub-alegações da hipótese H1. AR5 é um argumento contra ou a favor a hipótese H2.

Comparando a proposta de Kim et al. com o modelo IBIS e considerando os exemplos de uso apresentados por Kim et al, é possível notar que:

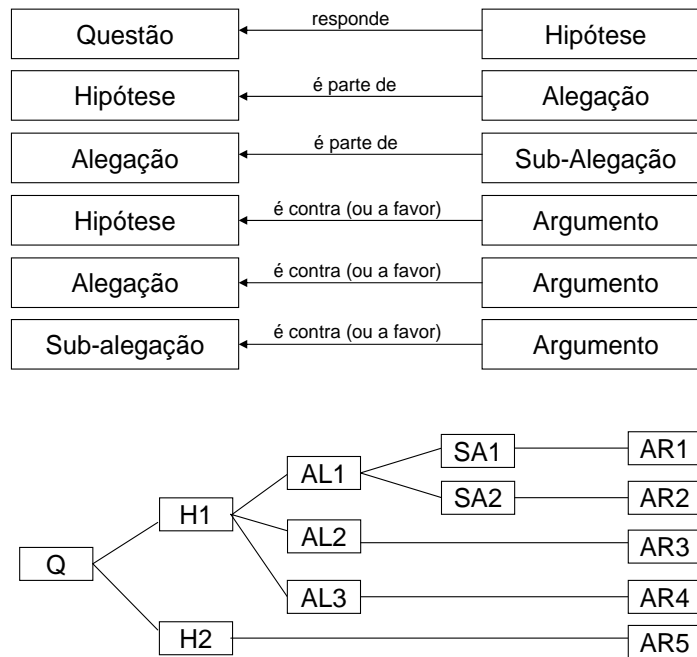


Figura 3.4: Representação de *design rationale* de acordo com a proposta de Kim et al. (1993)

- as entidades *questão* e *argumento* propostas no modelo IBIS foram adotadas no modelo para o contexto de pesquisa científica;
- há uma relação entre *hipótese* e *posição*, ou seja, em ambos os modelos, o objetivo é indicar possíveis soluções para uma questão gerada.

O elemento novo na proposta de Kim et al. são as alegações e sub-alegações, que podem ser vistas como entidades que favorecem o refinamento das hipóteses ou das alegações, ou seja, são utilizadas para tentar descrever a hipótese em partes. Os termos utilizados na elaboração da proposta são usuais em contexto de pesquisa.

Como aspecto positivo da proposta, pode ser mencionada sua simplicidade e o fato de estar fortemente fundamentada em um modelo de *design rationale* amplamente discutido. Como aspectos negativos da proposta, pode ser mencionado o fato de não ter sido apresentado como os autores obtiveram o modelo adaptado, ou seja, qual foi a motivação para o novo modelo e quais foram os requisitos e as atividades executadas para a elaboração da proposta. Outro problema observado é que resultados experimentais não foram indicados. É válido destacar que estas deficiências foram tratadas na proposta apresentada nesta tese.

3.6 Limitações, perspectivas e desafios em *design rationale*

Um dos resultados de uma pesquisa realizada recentemente, com oitenta e um projetistas de software, revelou que eles reconhecem a importância em documentar *design rationale* na prática e freqüentemente usam as informações capturadas (em diferentes formatos) para analisar as decisões tomadas durante o desenvolvimento dos projetos que participam (Tang et al., 2005). Além disso, eles indicaram duas principais barreiras para a documentação e o uso de *design rationale*:

1. falta de tempo para documentação das informações;
2. falta de padrões e ferramentas apropriadas para o processo que envolve a captura de *design rationale*.

Em relação à segunda barreira mencionada, ficou evidente na pesquisa realizada que os participantes gostariam que padrões e processos indicassem *porque, como, o que e quando* as informações de *design rationale* devem ser capturadas. Foi também obtido dos participantes que as ferramentas Microsoft (Word, Visio, Excel e Powerpoint) são geralmente utilizadas para a captura de *design rationale*, o que foi considerado um fator que os desmotiva para o uso da abordagem.

É importante notar que têm surgido iniciativas em relação ao tratamento de *design rationale* por padrões internacionais. O padrão IEEE Standard 1471-2000 (2000) propõe a associação de *design rationale* à descrição da arquitetura de software e de sistemas. No entanto, é apresentada somente uma definição para a abordagem e não são indicados quais os tipos específicos de informações devem ser capturadas. Em relação às ferramentas para *design rationale*, nota-se que alguns trabalhos estão sendo desenvolvidos no sentido de integrá-las a outras ferramentas comumente utilizadas no desenvolvimento de software, enfatizando as especificidades de cada fase do desenvolvimento (Canfora et al., 2000; Myers et al., 1999; van der Ven et al., 2006).

Uma tendência que pode ser observada em pesquisas atuais é o registro do *contexto* durante a captura de *design rationale*. O objetivo principal é promover a rastreabilidade, ou seja, fornecer informações que ajudem o projetista a entender, durante a recuperação de *design rationale*, o cenário no qual a captura das informações ocorreu (Breitling et al., 2006; Nguyen e Swatman, 2006). É possível notar também que as pesquisas envolvendo *design rationale* privilegiam a abordagem da argumentação ainda hoje. A maioria dos estudos tenta propor variações dos modelos de representação, em particular do modelo IBIS.

Haynes (2006) apresentou alguns desafios de pesquisa na área para os próximos anos. Segundo ele, é preciso investir esforços na captura ubíqua de *design rationale*, de forma a

evitar a interferência à progressão natural das atividades de projeto; na investigação sobre possibilidades de uso de *design rationale* como um veículo para facilitar a transferência de tecnologia entre organizações e na realização de estudos experimentais que contribuam para compor uma base de experiências relativas a *design rationale*.

3.7 Considerações finais

Neste capítulo foram apresentados conceitos e atividades relacionadas à adoção da abordagem de *design rationale*. O uso dessa abordagem em diferentes contextos também foi enfatizada.

Os principais conceitos apresentados neste capítulo que foram usados na elaboração da proposta apresentada nesta tese foram: (1) as abstrações e os relacionamentos dos modelos de argumentação IBIS, PHI e QOC, pois foi proposto um modelo de argumentação que contempla parcialmente os elementos desses modelos e (2) os mecanismos para a captura e a recuperação de *design rationale*, pois essas atividades foram priorizadas neste trabalho.

A partir do estudo realizado, foi possível conhecer as pesquisas que estão sendo realizadas mais recentemente na área e os principais desafios para os próximos anos. Além disso, essa revisão bibliográfica propiciou um referencial teórico sobre o qual os resultados apresentados nesta tese contribuem.

Concepção de um Processo para Projetos de Pesquisa em Software

4.1 Considerações iniciais

Para a elaboração do processo apresentado nesta tese, as atividades propostas por Humphrey (1995) em seu livro *A Discipline for Software Engineering*, Capítulo 13 (*Defining the Software Process*) foram executadas. Outras propostas foram procuradas na literatura, porém, para o propósito de *concepção de um processo*, o trabalho de Humphrey foi o único encontrado.

Na Seção 4.2 é apresentada uma descrição da abordagem proposta por Humphrey, que foi utilizada como fundamento teórico. Nas seções seguintes são apresentados os resultados da execução das cinco primeiras etapas. Os resultados da etapa de definição do processo são apresentados no Capítulo 5. Uma instanciação do processo, observado como uma atividade inicial da validação, é apresentado na Seção 5.5. A etapa 8 (melhorar o processo) não foi enfatizada neste trabalho.

4.2 Proposta de Humphrey para definição de processos de software

De acordo com Humphrey (1995), processos de software devem ser propostos quando não há abordagens apropriadas para as necessidades específicas dos usuários interessados. A definição de processos deve ser diretamente relacionada a “o que” o usuário precisa fazer, ou seja, deve estar relacionada aos seus requisitos. Em geral, processos são utilizados

quando o objetivo final é executar alguma tarefa repetitiva. Processos de software também são importantes no sentido de auxiliar a planejar e rastrear o trabalho, a guiar a execução e avaliação das tarefas e a propor melhorias. Humphrey sugeriu que um processo contenha os seguintes elementos:

- *Scripts* que descrevam como o processo deve ser colocado em prática;
- Formulários que auxiliem o registro dos dados que precisam ser armazenados;
- Padrões para guiar o trabalho e fornecer uma base para verificação da qualidade dos produtos e dos processos;
- Alternativas para a melhoria do processo de forma a assegurar que ele irá continuar a satisfazer as necessidades dos seus usuários, que evoluem constantemente.

A estratégia sugerida por Humphrey para a definição de processos de software é semelhante aos modelos convencionais para o desenvolvimento de produtos de software. Deve-se iniciar estabelecendo as necessidades dos usuários e terminar com a realização de testes seguidos pela liberação do processo. As atividades para a definição de um processo são:

1. **Determinar as características do processo:** envolve determinar a natureza dos produtos que serão obtidos a partir da execução do processo, identificar os principais atributos de cada produto e as características do processo necessárias para produzir tais atributos.
2. **Definir os objetivos do processo, as metas e os critérios de qualidade:** envolve identificar o que os usuários esperam do processo, os objetivos preliminares e as prioridades de melhoria do processo.
3. **Caracterizar o processo corrente:** para melhorar o processo já existente, é necessário compreender o processo corrente e, em seguida, melhorá-lo de forma incremental. É necessário descrever os principais passos, como estão relacionados, quais são os principais problemas observados pelos usuários, se há critérios de entrada e saída explícitos e se é possível rastrear e realizar medições com o processo.
4. **Caracterizar o processo alvo:** deve-se definir a estrutura e identificar os principais elementos do processo.
5. **Estabelecer uma estratégia de desenvolvimento do processo:** está relacionado à definição de “como” o processo será definido.

6. **Definir o processo inicial:** deve-se definir os *scripts*, os padrões e os formulários para o processo inicial. Este processo deve ser semelhante ao processo corrente, mas deve incluir mudanças que permitirão aproximar do processo alvo. Sugere-se que seja definido o processo como um todo, incluindo as diversas atividades. Em seguida, deve-se definir um sub-processo para cada atividade.
7. **Validar o processo inicial:** está relacionado ao teste do processo. Primeiramente, deve-se simular a execução do processo usando dados de um projeto que tenha sido desenvolvido recentemente. Se os dados necessários não estiverem disponíveis, deve-se considerar aqueles que for possível. Em seguida, o processo deve ser utilizado no desenvolvimento de um pequeno projeto ou no desenvolvimento de um protótipo.
8. **Melhorar o processo:** ao utilizar o processo, as omissões e os problemas se tornarão evidentes. As melhorias do processo se referem aos ajustes à primeira definição. À medida que os ajustes são feitos, a proposta converge para um processo inicial realístico.

4.3 Determinação das características do processo

Sobre a natureza dos produtos que o processo deverá produzir

Pretende-se elaborar um processo que seja útil para o desenvolvimento de projetos de pesquisa. Em termos de produção de software, o processo será utilizado para o desenvolvimento de protótipos, ou seja, não há o interesse no desenvolvimento de produtos de software¹. Além disso, o processo deverá ser útil para auxiliar a execução de atividades e a produção de artefatos comuns ao desenvolvimento de projetos de pesquisa, tais como a elaboração de um plano de pesquisa.

Os principais atributos do produto

Espera-se que os principais atributos dos produtos desenvolvidos utilizando o processo elaborado sejam simplicidade e manutenibilidade.

Sobre as características necessárias do processo para produzir os atributos do produto

A documentação compreensível e consistente das informações relevantes de um projeto é fundamental quando outras pessoas que não participaram do desenvolvimento precisam entender os artefatos produzidos (Schneidewind, 1987). Assim, considerando-se que a

¹No desenvolvimento de protótipos nos projetos de pesquisa, os objetivos estratégicos são de investigação e obtenção de conhecimento. No desenvolvimento de produtos, a preocupação com os objetivos de negócio são evidentes.

alta rotatividade dos membros e a identificação iterativa dos requisitos são características importantes do desenvolvimento dos projetos de pesquisa, o processo deverá enfatizar a documentação simples, compreensível e propensa a mudanças. Com isso, busca-se garantir que não somente o código, mas também a documentação, estejam disponíveis quando os estudantes e os pesquisadores deixam de fazer parte da equipe e outras pessoas passam a trabalhar na continuidade dos projetos.

4.4 Definição de objetivos, metas e critérios de qualidade do processo

De acordo com Humphrey (Humphrey, 1995), é fundamental identificar o que se espera obter com o processo que será projetado, ou seja, quais são as necessidades das pessoas que usarão o processo. De forma similar, Davenport (1994) observou que um dos primeiros passos na definição de um processo é explicitar quem são os seus usuários e o que eles esperam obter. Os autores ressaltam a importância em relação ao projeto de processos que satisfaçam de fato às exigências do(s) usuário(s).

De forma geral, o objetivo da proposta do processo está relacionado com: (1) a obtenção de melhorias na documentação de projetos de pesquisa envolvendo software, de forma a contribuir para a continuidade e evolução dos mesmos e (2) a organização das atividades executadas e artefatos gerados, de forma a contribuir para a reprodução de colaborações bem sucedidas.

Os seguintes usuários foram considerados para a definição do processo apresentado neste trabalho:

1. pesquisadores (estudantes e colaboradores);
2. coordenadores de projetos;
3. profissionais do setor industrial interessados em estabelecer parcerias com universidades ou centros de pesquisa.

Conforme apresentado na sub-seção seguinte, nesta etapa da concepção do processo buscou-se, primordialmente, identificar o que os usuários esperam do processo, ou seja, definir os requisitos da proposta.

Para a identificação dos requisitos dos usuários do processo foi realizada revisão da literatura. Não foi possível fazer a distinção entre requisitos dos pesquisadores e requisitos dos coordenadores dos projetos e, portanto, eles são apresentados indistintamente a seguir.

Usuários: pesquisadores e coordenadores de projetos

1. Requisito: *Atividades de pré-desenvolvimento*. Fonte: Ambati e Kishore (2004); Durscki et al. (2004); Hwang e Park (2006); Robillard e Robillard (1998); Walker (2003)

Atividades: definição do tema e planejamento do projeto de pesquisa, treinamento (no domínio da aplicação e outros temas de interesse dos pesquisadores)

2. Requisito: *Atividades de ciclo de vida*. Fonte: Ambati e Kishore (2004); Hwang e Park (2006); Robillard e Robillard (1998); Segal (2005)

Atividades: análise de requisitos, projeto, geração de código, testes

Observação: deve-se ressaltar também que, em relação às atividades de ciclo de vida, a combinação entre métodos ágeis e abordagens tradicionais, o uso do método da prototipação e do desenvolvimento iterativo também foram enfatizados por Segal (2005) e por Carlson (2002).

3. Requisito: *Atividades de gerenciamento*. Fonte: Ambati e Kishore (2004); Boldyreff et al. (2004); Chirouze et al. (2005); Conn (2004); Durscki et al. (2004); Hwang e Park (2006); Numprasertchai e Igel (2003); Robillard e Robillard (1998); Uwadia et al. (2006); Walker (2003)

Atividades: gerenciamento de tarefas, cronograma, custos, recursos, riscos, configuração e infra-estrutura

4. Requisito: *Atividades de garantia de qualidade de software*. Fonte: Alpern et al. (2005); Ambati e Kishore (2004); Cavin et al. (1989); Chirouze et al. (2005); Conn (2004); Hwang e Park (2006); Robillard e Robillard (1998); Walker (2003)

Atividades: verificação, validação, documentação, revisão técnica, controle de alterações, revisão por pares, medições em produtos e processos (envolve garantia de qualidade de artefatos de software e outros artefatos resultantes do trabalho de pesquisa)

5. Requisito: *Atividades de apoio ao desenvolvimento colaborativo*. Fonte: Alpern et al. (2005); Ambati e Kishore (2004); Boldyreff et al. (2004); Carlson (2002); Chirouze et al. (2005); Numprasertchai e Igel (2003); Rogers et al. (1998); Segal (2005); Succi e Zuliani (2004); Uwadia et al. (2006); Walker e Braude (1999)

Atividades: apoio à comunicação, definição de papéis, representação do processo cumprido e dos pontos de colaboração que devem ser compartilhados entre os

membros do grupo, integração de código e de outros artefatos, coordenação das atividades, registro do conhecimento comum ao grupo e percepção do grupo em relação ao contexto de trabalho

6. Requisito: *Atividades de apoio ao desenvolvimento distribuído*. Fonte: Alpern et al. (2005); Ambati e Kishore (2004); Bezroukov (1999); Chirouze et al. (2005); Perpich et al. (1997); Segal (2005); Succi e Zuliani (2004)

Atividades: apoio à comunicação, ao acesso ao conhecimento, à realização de atividades cooperativas, ao registro de experiências e discussões, à identificação de habilidades, experiências e recursos humanos, tecnológicos e econômicos de cada equipe; coordenação distribuída (por um coordenador local) e gerência distribuída do projeto, controle de artefatos (acompanhamento do desenvolvimento de artefatos) e resolução de problemas

7. Requisito: *Atividades de apoio ao desenvolvimento de pesquisas*. Fonte: Alpern et al. (2005); Athey e Plotnicki (2000); Boldyreff et al. (2004); Carlson (2002); Chirouze et al. (2005); Conn (2004); Segal (2005); Walker (2003)

Atividades: divulgação do projeto, *postmortem*, apoio à revisão da literatura, à realização de entrevistas e *surveys*, à execução de experimentos, à redação de documentos técnicos, à elaboração e disponibilização de material didático

Usuários: profissionais do setor industrial

1. Requisito: *Atividades de apoio à transferência de conhecimento* (boas práticas de Engenharia de Software). Fonte: Beckman e Khajenoori (1997); Jenkins (2004); Kornecki et al. (2003); Rogers et al. (1998); Rost (2005)

Atividades: elaboração de material para treinamento, seleção de parceiro, negociação de suporte financeiro para universidade e bolsa de estudos para estudantes envolvidos, planejamento de estágio de estudantes e pesquisadores na indústria, planejamento de estágio de profissionais na universidade, criação de infra-estrutura na universidade, transferência de soluções e tecnologias para o parceiro da indústria, elaboração de plano de operação (detalhes do projeto de pesquisa, métodos, pessoas, técnicas e artefatos a serem produzidos), definição de papéis e responsabilidades, submissão de proposta à agência financiadora, avaliações dos parceiros envolvidos em relação aos resultados obtidos, captura e reuso de conhecimento

2. Requisito: *Atividades de apoio à formação de parcerias universidade-indústria*. Fonte: Chang e Trubow (1990); Ellis et al. (2003); Hwang e Park (2006); Kornecki et al. (2003); Macke et al. (1996); Walker e Braude (1999)

Atividades: definição de objetivos claros, identificação dos benefícios que cada participante terá com a parceria, planejamento e gerenciamento do projeto, apoio à comunicação entre os participantes, estabelecimento de acordo sobre confidencialidade de informações, seleção de informações que serão disponibilizadas na parceria, negociação de suporte financeiro, identificação de recursos fornecidos por cada uma das partes envolvidas, geração de relatórios e documentos pelo pesquisador responsável sobre o funcionamento de laboratórios, relações entre academia e indústria e elaboração de contratos incluindo direitos de propriedade do software (*copyrights* e patentes), direito de publicação dos resultados obtidos, propriedade intelectual, divisão de lucros, controle da especificação dos projetos

Em síntese, os requisitos observados na literatura foram divididos em 9 áreas: atividades de pré-desenvolvimento, de ciclo de vida, de gerenciamento, de garantia de qualidade de software, de apoio ao desenvolvimento colaborativo, de apoio ao desenvolvimento distribuído, de apoio ao desenvolvimento de pesquisas, de apoio à transferência de conhecimento e de apoio à formação de parcerias universidade-indústria. Nota-se que as áreas cobertas pela norma ISO/IEC 12207 (1995) foram abordadas pelos pesquisadores, o que motivou a redefinição dos processos apresentados na norma para o contexto de desenvolvimento de pesquisas. Além disso, houve a inserção de novos processos, para atender ao domínio do desenvolvimento de pesquisas.

Sobre os critérios de qualidade do processo

De acordo com a proposta de Humphrey, é fundamental definir os critérios de qualidade do processo que deverão servir como base para permitir caracterizar o que é um bom ou um mau processo. O autor propõe que, primordialmente, um processo possa ser entendido e usado facilmente. Estes atributos foram considerados como critérios de qualidade para o processo proposto. Assim, foram definidas métricas que pudessem ser utilizadas para avaliá-los.

Os atributos e as métricas apresentadas a seguir foram adaptações da norma ISO/IEC 9126 (2002), que trata da avaliação de produtos de software, para o contexto de avaliação de processos. As métricas não foram coletadas no contexto da proposta apresentada nesta tese.

1. Facilidade de entendimento

Definição do atributo: facilidade com que as descrições do processo podem ser entendidas.

Proposta de avaliação: o processo é fácil de entender?

Origem: norma ISO/IEC 9126, parte 2 (métricas externas), característica usabilidade, subcaracterística intelegibilidade.

Método de aplicação e métricas: Este atributo pode ser avaliado executando-se três tarefas, conforme descritas a seguir.

Tarefa 1: treine usuários e faça entrevistas ou observe seus comportamentos. Conte o número de atividades do processo que foram compreendidas adequadamente e compare com o número total de atividades consideradas no treinamento.

- **Métrica:** $X = A/NA$

A = número de atividades no treinamento cuja proposta é compreendida corretamente pelo usuário; NA = número total de atividades; X é um valor absoluto.

- **Interpretação do resultado:** $0 \leq X \leq 1$.

O valor é considerado melhor quanto mais próximo X estiver de 1.

Tarefa 2: treine usuários e faça entrevistas ou observe seus comportamentos. Conte o número de atividades cujas entradas e saídas foram compreendidas corretamente e compare com o número total de atividades consideradas no treinamento.

- **Métrica:** $Y = B/NA$

B = número de atividades no treinamento cujas entradas e saídas foram compreendidas corretamente; NA = número total de atividades; Y é um valor absoluto.

- **Interpretação do resultado:** $0 \leq Y \leq 1$.

O valor é considerado melhor quanto mais próximo Y estiver de 1.

Tarefa 3: realize experimentação com o processo após o treinamento com o usuário. Conte o número de atividades executadas corretamente e compare com o número total de atividades que fizeram parte do experimento.

- **Métrica:** $Z = C/NA$

C = número de atividades completadas com sucesso na experimentação do processo; NA = número total de atividades; Z é um valor absoluto.

- **Interpretação do resultado:** $0 \leq Z \leq 1$.

O valor é considerado melhor quanto mais próximo Z estiver de 1.

2. Facilidade de uso

Definição do atributo: facilidade com que o processo pode ser utilizado.

Proposta de avaliação: o processo é fácil de usar?

Origem: norma ISO/IEC 9126, parte 2 (métricas externas), característica usabilidade, subcaracterística operacionalidade.

Método de aplicação e métricas: este atributo pode ser avaliado executando-se 3 tarefas, conforme descritas a seguir.

Tarefa 1: realize experimentação com o processo após o treinamento com o usuário. Faça entrevistas com foco na consistência da descrição do processo. Conte o número de atividades do processo nas quais o usuário encontra inconsistências inaceitáveis e compare com o número total de atividades consideradas no experimento.

- **Métrica:** $X = 1 - A/NA$

A = número de atividades do processo nas quais o usuário encontra inconsistências inaceitáveis; NA = número total de atividades; X é um valor absoluto.

- **Interpretação do resultado:** $0 \leq X \leq 1$.

O valor é considerado melhor quanto mais próximo X estiver de 1.

Tarefa 2: realize atividades de experimentação do processo após o treinamento e observe o comportamento do usuário. Conte o número de vezes que o usuário pausa por um longo período (os avaliadores definem o que consideram um longo período) ou falha repetidamente na mesma operação devido a dificuldades na compreensão da descrição do processo e compare com o número total de atividades consideradas no experimento.

- **Métrica:** $Y = B/TO$

B = número de vezes o usuário pausa por um longo período ou falha repetidamente na mesma operação devido a dificuldades na compreensão das mensagens; TO = tempo de observação; Y é um valor do tipo razão.

- **Interpretação do resultado:** $Y \geq 0$.

O valor é considerado melhor quanto mais próximo Y estiver de 0.

Tarefa 3: realize atividades de experimentação do processo após o treinamento e observe o comportamento do usuário. Conte o número de atividades que o usuário customiza de forma bem sucedida e o número de tentativas de customização e compare com o número total de atividades consideradas no experimento.

- **Métrica:** $Z = C/TC$

C = número atividades customizadas com sucesso; TC = número de tentativas de customização; Z é um valor absoluto.

- **Interpretação do resultado:** $0 \leq Z \leq 1$.

O valor é considerado melhor quanto mais próximo Z estiver de 1.

4.5 Caracterização do processo corrente

De acordo com Humphrey (Humphrey, 1995), as chances de se conseguir resultados positivos são maiores quando há planos para a realização de várias mudanças incrementais em um processo já existente, ao invés de uma única mudança drástica. Assim, não é viável definir um processo completamente novo em que sejam desconsideradas as práticas adotadas pela comunidade que executa o processo. O autor sugere, portanto, que se conheça o processo que é executado e que se deseje melhorar. Para isso, alguns pontos principais precisam ser avaliados, ou seja, é preciso descrever as principais atividades do processo atual, como estão relacionadas, o tempo gasto para executar cada uma delas, os problemas encontrados, os critérios de entrada e saída e os planejamentos e as medições que são realizadas com a execução do processo atual.

A abordagem utilizada neste trabalho foi a de analisar práticas que foram executadas durante o desenvolvimento de projetos de pesquisa e que foram registradas em artigos científicos. Foi possível identificar somente práticas de desenvolvimento de software, e não práticas mais genéricas de desenvolvimento de projetos de pesquisa. Foram analisados documentos de 36 projetos que (1) incluíram produção de software, (2) foram desenvolvidos em diferentes países e (3) cujas publicações foram reportadas na literatura nos últimos cinco anos. Resumos foram evitados, por fornecerem apenas uma descrição breve do projeto. Os documentos foram buscados em periódicos, anais, teses, dissertações e repositórios de projetos na internet. É importante observar que uma ameaça à validade dos resultados obtidos se refere ao fato de que os documentos analisados não tinham como objetivo descrever o processo utilizado e, portanto, é possível que somente uma parte das atividades executadas tenham sido apresentadas. Além disso, não foi possível inferir o quanto os processos foram executados. Os resultados obtidos são apresentados na Figura 4.1 em termos de porcentagem da execução das práticas em relação ao número total de projetos avaliados. No eixo horizontal são apresentadas as atividades mais indicadas nos documentos analisados e no eixo vertical são apresentadas as porcentagens. A relação entre os projetos analisados e as práticas executadas é apresentada no Apêndice 1.

Pelo resultado apresentado no gráfico é possível notar que, por se tratarem de projetos de software, 100% dos projetos mencionaram implementação. As atividades executadas por pelo menos 50% dos projetos foram análise de requisitos e projeto. O resultado obtido foi utilizado na elaboração do processo proposto neste trabalho de duas formas:

1. balancear requisitos dos usuários e atividades conhecidas por eles para a elaboração do processo e
2. colaborar na indicação de atividades em que estudos adicionais sobre melhorias na documentação (tendo em vista a melhoria da manutenção dos projetos) seriam realizados, considerando a abordagem de *design rationale*.

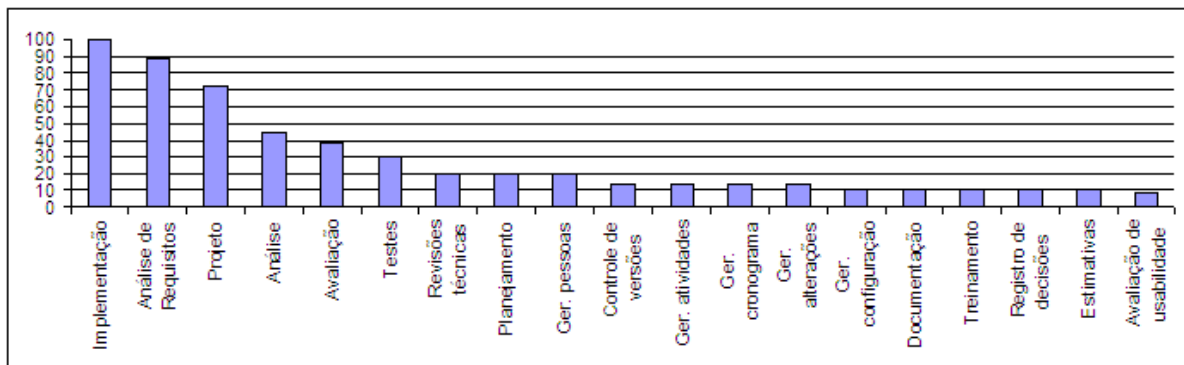


Figura 4.1: Práticas executadas no desenvolvimento de 36 projetos de pesquisa (em termos de porcentagem)

Humphrey sugere que a melhoria de um processo seja iniciada por atividades que são conhecidas pelos usuários do processo. Rocha et al. (2001a) apresentaram um exemplo de definição de processos em que essa estratégia foi utilizada na prática e os resultados obtidos foram positivos.

Uma revisão da literatura foi conduzida, com o objetivo de coletar dados sobre as práticas genéricas de desenvolvimento de projetos de pesquisa envolvendo software (atividades além daquelas de ciclo de vida de software). Em resumo, foi possível observar que:

- poucas atividades referentes a planejamento e controle são executadas, pois há muitas incertezas em relação ao desenvolvimento de projetos de pesquisa. Há incertezas, por exemplo, em relação a: (1) o tempo gasto para provar uma teoria ou obter resultados válidos para experimentos e (2) os requisitos que serão considerados no desenvolvimento do projeto, pois eles são identificados gradualmente, à medida em que o projeto é desenvolvido e os resultados são percebidos (Huljenic et al., 2005);
- a documentação gerada nos projetos de pesquisa não é ideal pois, em geral, está restrita a documentos científicos, como artigos, dissertações e teses (Boldyreff et al., 2004; Walker, 2003);
- há alta rotatividade dos membros que participam do desenvolvimento de projetos de pesquisas e nem sempre a documentação disponível é adequada para que novos participantes possam entender o estágio em que o projeto se encontra (Cavin et al., 1989; Igel e Numprasertchai, 2004);
- as atividades relacionadas a treinamento geralmente ocorrem de formas diferentes, de acordo com o contexto em que é realizado. Em geral, o treinamento em ambiente acadêmico ocorre por meio de uma combinação de aulas e tutorias e não há envolvimento em termos práticos dos participantes com o conteúdo apresentado.

No contexto industrial, é comum que aulas breves sejam ministradas e trabalhos práticos sejam desenvolvidos por pequenos grupos (Bleek et al., 2005).

Durante a etapa de caracterização do processo corrente, Humphrey ainda sugere que algumas questões sejam respondidas. Para responder as questões, foram considerados, principalmente, os dados apresentados nesta seção. As respostas representam também um resumo sobre as características do desenvolvimento de projetos de pesquisa apresentadas em outras seções.

1. *É possível descrever as principais atividades do processo corrente, como estão relacionadas e o tempo gasto para execução de cada uma delas?*

De forma geral observou-se que são executadas principalmente atividades de ciclo de vida de desenvolvimento de software, destacando-se: implementação, análise de requisitos e projeto. Não foram encontradas informações sobre o relacionamento entre as atividades e o tempo gasto para executá-las.

2. *Há problemas relacionados aos processos correntes? Liste-os.*

Há dificuldades na manutenção dos projetos. É comum a re-escrita (ao invés da evolução) de funcionalidades dos protótipos. Boldyreff et al. (2004) e Walker (2003) destacaram as limitações na documentação gerada nos projetos, que é um dos principais fatores que dificultam a continuidade dos mesmos. Problemas com o gerenciamento de projetos, principalmente quando são desenvolvidos de forma distribuída, foram indicados por Cavalcanti et al. (2002) e Uwadia et al. (2006). Observam-se também problemas no estabelecimento de parcerias entre indústria e academia, devido a dificuldades na definição de objetivos e de papéis e no gerenciamento de recursos (Kornecki et al., 2003).

3. *Os passos do processo corrente possuem artefatos de entrada e de saída bem definidos?*

Não foram encontrados registros que pudessem indicar tais critérios.

4. *O processo corrente é planejado e rastreado?*

Não. Conforme apresentado por Huljenic et al. (2005), na prática, poucas atividades de planejamento de projetos de pesquisa são realizadas ou, pelo menos, reportadas. Os autores consideram também que as diferenças entre o desenvolvimento de projetos de pesquisa e o desenvolvimento de software na indústria impõem que o gerenciamento dos projetos ocorra de forma diferenciada em ambos os casos.

5. *São realizadas medições suficientemente no processo corrente de forma a possibilitar que planos de melhoria sejam propostos?*

Foram encontrados alguns relatos que indicam interesse da comunidade científica em medir os processos utilizados nos laboratórios de pesquisa. Coccia (2001, 2003)

propôs um modelo, aplicado em duzentos institutos, para coleta de dados sobre desempenho e produtividade em laboratórios de pesquisa. As métricas estabelecidas no modelo estão relacionadas a: (1) *auto financiamento*, que é a quantidade de recursos obtidos a partir da parceria entre universidade e indústria; (2) *treinamento*, que é a quantidade de pessoas treinadas no instituto, por exemplo, número de pessoas que obtiveram alguma titulação e número de pessoas que participaram de cursos temporários; (3) *ensino*, que é a quantidade de cursos oferecidos no instituto e (4) *produção científica*, que é a quantidade de artigos publicados em periódicos nacionais e internacionais. Outras iniciativas e experiências também foram apresentadas (Harris e Kaine, 1994; Luwel et al., 1999), no entanto, não foi discutida a relação entre a aplicação de métricas e melhoria dos processos. Em outros estudos, a ênfase está no processo de avaliação utilizado e há pouca discussão sobre as medições realizadas (Miyoshi, 1998; Walker, 2003). Nestes casos, são apresentadas relações entre as avaliações e a melhoria dos processos. De forma geral, nota-se que há iniciativas no sentido de medir, avaliar e melhorar os processos utilizados no desenvolvimento de pesquisas. Não foram encontradas, porém, generalizações sobre medições que pudessem indicar como possíveis planos de melhoria podem ser implantados.

4.6 Caracterização do processo almejado

De acordo com Humphrey (Humphrey, 1995), nesta etapa deve-se identificar os objetivos a serem alcançados em relação ao processo que está sendo proposto, bem como a estrutura e os principais elementos do processo.

Para a apresentação do processo padrão foi utilizada a estrutura indicada na norma ISO/IEC 12207, em que os processos são divididos em processos fundamentais, processos de apoio e processos organizacionais. Assim, a estrutura geral do processo almejado pôde ser então definida:

- 1. Processos Fundamentais:** Processo de aquisição, Processo de iniciação, Processo de desenvolvimento, Processo de operação, Processo de manutenção.
- 2. Processos de Apoio:** Processo de documentação, Processo de gerência de configuração, Processo de garantia da qualidade, Processo de verificação, Processo de validação, Processo de revisão, Processo de resolução de problemas, Processo de revisão sistemática, Processo de preparação de documentos científicos, Processo de elaboração de módulos educacionais, Processo de *postmortem* e Processo de transferência tecnológica.
- 3. Processos Organizacionais:** Processo de treinamento, Processo de gerência, Processo de infra-estrutura, Processo de melhoria, Processo de planejamento, Processo

de divulgação, Processo de comunicação, Processo de coordenação e Processo de estabelecimento de parceria universidade-empresa.

4.7 Estabelecimento de uma estratégia de desenvolvimento do processo

De acordo com Humphrey (Humphrey, 1995), nesta etapa a(s) pessoa(s) responsável(is) pela definição do processo deve(m) definir e executar tarefas que a(s) auxiliem a conhecer os processos. Algumas sugestões fornecidas para a definição da estratégia são: (1) para processos que não são familiares, observe como são executados ou colete dados que os tornem explícitos; (2) especifique os processos incluindo passos, atividades ou tarefas que, por experiência, tenham sido executados com sucesso; (3) se não for possível compreender o processo antes de sua especificação, adote uma abordagem com foco na simplicidade.

Neste trabalho, a estratégia apresentada a seguir foi utilizada para a estruturação do processo para desenvolvimento de projetos de pesquisa envolvendo software.

1. De forma geral, para a apresentação do processo foi utilizada a abordagem definida por Maidantchik (1999), que propõe a definição de um processo padrão, a partir do qual são definidos processos especializados. Para os diferentes projetos a serem desenvolvidos são realizadas instanciações dos processos especializados (Sub-seção 4.7.1).
2. Foram realizados três estudos de caso com o objetivo de adquirir conhecimento sobre os processos de documentação, manutenção e comunicação no domínio de pesquisa (Sub-seção 4.7.2).
3. Foi realizado outro estudo de caso em que foi avaliado o processo utilizado no desenvolvimento do projeto de uma aplicação *web* denominada *Agenda No-Risk Planning* (Moura, 2001; Ribeiro, 2003) com o objetivo de conhecer como os estudantes e os pesquisadores envolvidos trataram questões relacionadas ao processo de desenvolvimento em um projeto real, em que havia rotatividade das pessoas envolvidas e em que os usuários finais do sistema solicitavam manutenções constantemente. Além disso, iniciativas de melhoria do processo foram consideradas e foi possível observar como alguns processos descritos no contexto de processos fundamentais, processos organizacionais e processos de apoio poderiam ser melhorados e qual foi o impacto da melhoria no projeto (Sub-seção 4.7.3).
4. Em relação aos processos para os quais não foram coletados dados que pudessem auxiliar nas suas especificações, buscou-se apresentá-los de forma simples, conforme sugerido por Humphrey. Para isso, foi utilizada uma estratégia semelhante àquela

usada na descrição do modelo CMMI (2006), em que exemplos para as práticas base são fornecidos. Neste trabalho, foram apresentados exemplos, referências adicionais, comentários e lições aprendidas indicadas por pesquisadores que tiveram alguma experiência com a execução do processo.

A seguir são descritos os três primeiros itens da estratégia utilizada.

4.7.1 Abordagem para apresentação do processo

A abordagem apresentada por Maidantchik (1999) para a definição de processos sugere a proposição de um processo padrão, a especialização do processo padrão e a instanciação do processo especializado para projetos específicos, conforme apresentado na Figura 4.2. A proposta apresentada por Rocha foi utilizada inicialmente para o contexto da definição de um processo de software para equipes geograficamente dispersas e para processo de software orientado a domínio (Rocha et al., 2001a). Posteriormente foi utilizado para a definição de um processo para desenvolvimento de módulos educacionais (Barbosa, 2004).

O processo padrão tem como principal objetivo fornecer uma estrutura única que será utilizada pela organização em seus projetos de software, constituindo a base para a definição de seus processos. A especialização corresponde ao processo no qual uma abstração é representada de forma específica a um determinado contexto, por meio da inclusão, modificação ou encapsulamento de seus atributos. É necessário ter como referência um modelo que determine os diferentes níveis de maturidade de um processo e como alcançá-los. O processo especializado deve ser instanciado para ser utilizado em um projeto específico, de forma a atender suas características (Rocha et al., 2001a). Na Figura 4.2 a abordagem proposta é esquematizada no contexto da definição de um processo para equipes geograficamente dispersas.

Neste trabalho, para a definição do processo padrão (Apêndice B) foram consideradas a norma ISO/IEC 12207 (1995) e os objetivos, requisitos, critérios e dificuldades discutidos neste capítulo. Para a especialização do processo foi considerado o modelo de maturidade CMMI (2006) (Seção 5.4). Para a instanciação do processo, foi apresentado um exemplo enfatizando a alocação de métodos, técnicas de desenvolvimento, recursos humanos e tecnológicos para as atividades.

4.7.2 Estudos de caso: documentação, manutenção e comunicação

Os três estudos apresentados a seguir foram realizados no contexto de disciplinas de cursos de graduação do ICMC-USP (estudos de caso 1 e 3) e da Universidade Federal de São Carlos (estudo de caso 2). O objetivo foi observar como os estudantes lidam com determinados processos, simulando um ambiente de desenvolvimento de pesquisas: a maioria dos estudantes não tinha experiência no domínio de aplicação proposto e

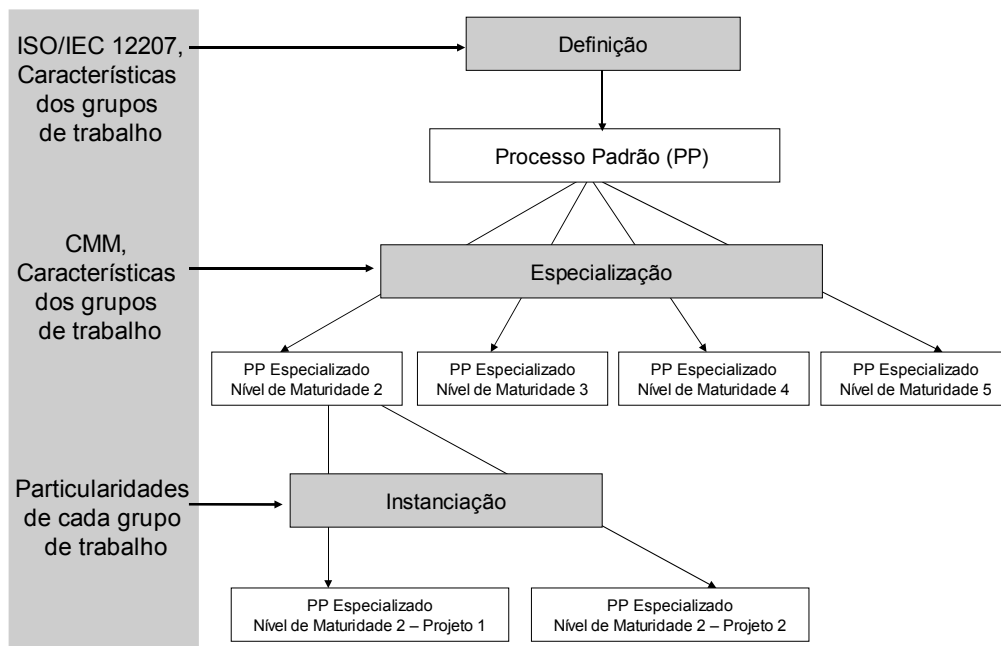


Figura 4.2: Definição de processo de software para equipes geograficamente distribuídas (Rocha et al., 2001a)

possuía experiência acadêmica em desenvolvimento de projetos de software; os trabalhos propostos foram desenvolvidos em grupos; houve rotatividade de pessoas para cumprir as tarefas planejadas, ou seja, um grupo iniciava e outro grupo finalizava as atividades e, em particular no estudo de caso 3, foi solicitado que simulassem um ambiente em que os membros da equipe estivessem geograficamente dispersos e utilizassem alternativas às reuniões presenciais para comunicação.

A seguir é apresentada uma breve descrição dos estudos e a indicação sobre a utilização dos resultados no processo proposto neste trabalho.

Estudo de caso 1: foco na melhoria da documentação de projetos de software

Foi realizado um estudo de caso no segundo semestre de 2002 com o objetivo de investigar se a abordagem de *design rationale* poderia contribuir para melhorar a documentação de projetos de software. Alguns resultados já haviam sido apresentados na literatura, porém o foco estava em outras áreas de conhecimento, por exemplo, Engenharia Mecânica (Karsenty, 1996). O objetivo foi, portanto, obter indícios sobre a aplicabilidade de *design rationale* no contexto de software. Foram investigados dois pontos principais:

1. se *design rationale* contribui para que engenheiros de software compreendam um projeto anterior;
2. se engenheiros de software estão interessados em *design rationale* quando precisam compreender um projeto.

Treze estudantes foram divididos em grupos e foi solicitado que cumprissem determinadas fases de um ciclo de vida de software tradicional para o desenvolvimento de um projeto, ou seja, análise de requisitos, projeto, projeto de interfaces e testes. Ao mesmo tempo, foi solicitado que capturassem e registrassem *design rationale* utilizando o modelo IBIS. Os projetos foram trocados entre os grupos (os temas eram diferentes para cada grupo) e foi solicitado que implementassem o protótipo considerando os resultados das fases cumpridas anteriormente, por outro grupo.

Foram elaboradas duas questões para auxiliar a análise dos resultados. A primeira questão foi “para cada estágio do ciclo de vida, as informações de *design rationale* foram úteis para auxiliar a compreensão do projeto?”. Considerando todas as fases do projeto, foi observado que, em média, *design rationale* foi importante por ajudar a responder 34,99% das dúvidas dos estudantes quando eles precisaram compreender o projeto desenvolvido por outra equipe. Os resultados em relação a cada um dos estágios do ciclo de vida indicou a utilidade de *design rationale* na compreensão das fases iniciais do desenvolvimento do projeto, ou seja, análise de requisitos e projeto arquitetural. Resultados semelhantes foram obtidos para outras áreas de conhecimento (Karsenty, 1996) e para o setor industrial (Conklin e Burgess-Yakemovic, 1996).

A segunda questão foi “os engenheiros de software estão interessados em *design rationale*?”. Foi observada a frequência de acesso às informações de *design rationale* pelos grupos. Os resultados obtidos não foram conclusivos, pois nem todos os grupos registraram os dados conforme solicitado, ou seja, foi registrado apenas que o acesso às informações ocorreu. Foi observado que todas as informações de *design rationale* disponibilizadas foram consultadas. Alguns estudantes mencionaram a importância em especializar o modelo de *design rationale* para as diferentes fases do ciclo de vida. A descrição e os resultados do estudo de caso foram apresentados em Paiva e Fortes (2005).

Os resultados obtidos com a realização deste estudo de caso indicaram a importância de *design rationale* no contexto da melhoria da documentação de software, quando há rotatividade entre as pessoas que desenvolve um projeto. Em particular, foram observados benefícios quando os artefatos gerados na fase de análise de requisitos precisaram ser compreendidos para que o desenvolvimento do protótipo pudesse ser realizado. Este resultado motivou a realização de estudos adicionais sobre o uso de *design rationale* nas fases de análise de requisitos e projeto de software, com o objetivo de melhorar a documentação obtida. É importante ressaltar que outra motivação foi o resultado apresentado na Figura 4.1 em relação ao processo executado atualmente no desenvolvimento de projetos de pesquisa. Foi observado que as fases de análise de requisitos e projeto estão entre as três mais valorizadas. Assim, observou-se que priorizá-las seria interessante, pois agregaria valor a uma atividade que é reconhecida pelos pesquisadores e que apresentou resultados positivos em relação ao uso de *design rationale*.

Estudo de caso 2: foco na importância de *design rationale* para manutenção de software

Este estudo foi realizado por doze estudantes de pós-graduação no segundo semestre de 2003 com o objetivo de investigar:

1. a importância de *design rationale*, resultante de um processo de reengenharia, na manutenção de um sistema de software;
2. a importância de *design rationale* na redução de tempo gasto na manutenção de software. Este segundo item está relacionado a este trabalho.

Os estudantes foram divididos em 4 grupos e foi proposta uma atividade de manutenção em um sistema de oficina eletrônica, composta por duas fases. Na primeira fase, foi solicitado que o sistema permitisse a realização de uma consulta adicional. Na segunda fase foi solicitada a inclusão de uma funcionalidade. Para a realização da primeira manutenção, os grupos 1 e 2 tiveram acesso às informações de *design rationale* e para a segunda manutenção, apenas os grupos 3 e 4 tiveram acesso. Os estudantes tiveram liberdade para estipular seus horários para desenvolvimento do trabalho, porém, foi solicitado que registrassem em um planilha fornecida o tempo gasto em cada uma das atividades.

Apesar de ser um estudo de caso piloto, foi observado que as informações de *design rationale* documentadas durante a reengenharia do sistema foram importante auxílio para o entendimento do projeto durante a realização da atividade de manutenção. Além disso, o resultado obtido indicou que o tempo gasto na manutenção foi menor quando as informações de *design rationale* estavam disponíveis. A descrição e os resultados do estudo de caso foram apresentados em Cagnin et al. (2004).

De forma semelhante ao resultado obtido com a realização do estudo de caso 1, o resultado obtido neste estudo motivou a realização de estudos adicionais envolvendo *design rationale*, tendo em vista a melhoria na manutenção dos projetos.

Estudo de caso 3: foco na comunicação entre os participantes do projeto

Para entender como os estudantes lidam com as questões relacionadas à comunicação quando um projeto distribuído é desenvolvido, foi realizado um estudo de caso no segundo semestre de 2004 em que o principal objetivo era tornar explícitas as características do processo da comunicação, que é um requisito dos usuários do processo proposto.

Como atividade proposta, foi solicitado que estudantes de duas turmas da disciplina de Engenharia de Software do ICMC-USP (94 estudantes no total) realizassem determinadas alterações (pré-definidas) em código fonte desenvolvido por estudantes de outra disciplina, utilizando ferramentas de software livre para apoiar a comunicação e simulando um ambiente de desenvolvimento distribuído de software. Os estudantes foram incentivados a desenvolver os projetos de forma colaborativa, dividindo tarefas e registrando dúvidas,

comentários e decisões tomadas, sem realizar encontros presenciais. Eles estavam matriculados em dois cursos diferentes e os grupos foram formados por estudantes de turmas diferentes.

Como resultados obtidos, em resumo, destacam-se:

1. no início do desenvolvimento do projeto foi notada a resistência dos estudantes em relação ao estabelecimento inicial da comunicação, o que indicou a necessidade de reforçar o treinamento (em termos teóricos e práticos) e motivar a equipe em relação à comunicação;
2. quando as dúvidas em relação à resolução do problema começaram a surgir, a comunicação foi intensificada e aos poucos se tornou mais natural;
3. os estudantes se empenharam em conhecer novas ferramentas que auxiliassem a comunicação e a vencer as dificuldades que surgem quando a distribuição geográfica é uma característica do desenvolvimento.

A descrição e os resultados do estudo de caso foram apresentados em Paiva et al. (2005). Os resultados obtidos foram positivos no sentido de indicar que é possível vencer as limitações do ambiente de desenvolvimento de pesquisas, como a escassez de recursos, quando há motivação dos estudantes em relação à comunicação e ao desenvolvimento distribuído. A importância do treinamento também foi reforçada.

4.7.3 Estudo de caso: avaliação e melhoria do processo utilizado no projeto *No-Risk Planning*

O projeto *No-Risk Planning* foi iniciado em 2001 por um estudante do curso graduação em Ciência da Computação do ICMC-USP como projeto de fim de curso (Moura, 2001). O objetivo foi desenvolver um sistema de agenda que estivesse disponível na *web* e permitisse o agendamento de compromissos de grupos de trabalho. Durante o desenvolvimento do protótipo inicial, apenas o código fonte e o modelo lógico da base de dados foram disponibilizados.

Tendo em vista as dificuldades enfrentadas por dois estudantes (um aluno de iniciação científica e um de mestrado) em dar continuidade ao desenvolvimento do projeto (Ribeiro, 2003), decidiu-se dar ênfase ao processo de desenvolvimento utilizado, que era composto, inicialmente, pelas atividades de modelagem do banco de dados e de codificação. Dessa forma, em paralelo à evolução das funcionalidades do software, iniciaram-se as atividades de melhoria do processo de desenvolvimento do sistema. O modelo IDEAL (Gremba e Myers, 1997) e a norma ISO/IEC 12207 (1995) foram utilizados para guiar a execução das atividades de melhoria do processo (Paiva et al., 2003).

Em uma primeira iniciativa, foram priorizadas, basicamente, atividades de ciclo de vida (processos fundamentais), atividades de gerência (processos organizacionais) e atividades

de gerência de configuração (processos de apoio). Em relação aos processos fundamentais, foram propostas melhorias na execução das atividades de especificação de requisitos, análise do sistema, projeto navegacional do sistema, realização de testes de unidade e de aceitação. Para os processos organizacionais foram considerados o gerenciamento de prazos e de recursos. As atividades relacionadas aos processos de apoio foram o controle de versões e a garantia da conclusão, consistência e correção dos itens. A equipe responsável pelo projeto se empenhou em executar as atividades priorizadas e, como resultado, um processo de desenvolvimento de software inicial foi estabelecido e a documentação do sistema foi notoriamente melhorada (Ribeiro et al., 2003).

A partir do uso expressivo do sistema por usuários finais, conforme apresentado por Freire et al. (2004), diversos requisitos novos surgiram e notou-se que a implementação eficiente desses requisitos exigia mais melhorias em termos de documentação e melhor controle (gerenciamento) do processo de desenvolvimento. Dessa forma, processos especializados para desenvolvimento de software para ambiente *web* foram investigados, tendo sido adotada a proposta elaborada por Pressman (2001), que propõe que sejam executadas as atividades de formulação e planejamento, análise, engenharia, geração de testes e avaliação com usuários; além de considerar princípios de gerenciamento. Com a adoção dessa abordagem foi possível considerar no processo elementos característicos de um desenvolvimento voltado para o ambiente *web*, por exemplo, a evolução contínua das aplicações e o planejamento de interfaces considerando elementos do contexto *web* (Fortes et al., 2004a).

É importante destacar também que o desenvolvimento do projeto *No-Risk Planning* passou a ser realizado utilizando o conjunto de ferramentas oferecidas pela Incubadora Virtual da Fapesp, ou seja, foram utilizadas ferramentas para apoio às atividades de controle de versões, controle de bugs, gerenciamento de tarefas, requisições de novas funcionalidades, gerenciamento de documentação, listas de discussão, fóruns, entre outros. A utilização dessas ferramentas também contribuiu para a melhoria do processo. Outras informações sobre a realização do estudo de caso podem ser encontradas em Paiva et al. (2004).

A experiência adquirida com a realização deste estudo de caso foi importante principalmente em relação à forma de implantação do processo, ou seja, foi observada na prática a importância em propor a melhoria gradual de um processo e em considerar elementos do domínio e do tipo de aplicação desenvolvido para que os esforços de melhoria sejam bem sucedidos. Além disso, os resultados obtidos com o uso de ferramentas que apóiam o desenvolvimento de projetos distribuídos em ambiente de pesquisa foram considerados.

As etapas seguintes da proposta de Humphrey para definição de processos se referem à definição e à validação do processo. Os resultados obtidos com a execução destas etapas são apresentados no Capítulo 5.

4.8 Considerações finais

Neste capítulo foram apresentadas as atividades executadas para a concepção de um processo para o desenvolvimento de projetos de pesquisa. A abordagem proposta por Humphrey (1995) para a definição de processos de software foi utilizada. O principal benefício obtido foi a possibilidade de tomar consciência e refletir sobre diversos elementos fundamentais da definição de um processo. Provavelmente a importância desses elementos não teria sido observada se uma abordagem sistemática não tivesse sido utilizada.

Em particular, duas limitações podem ser destacadas em relação à concepção do processo apresentada neste capítulo. A primeira se refere à fase de definição de objetivos e metas. De acordo com a proposta de Humphrey, nesta fase devem ser definidas métricas e os resultados esperados devem ser indicados. Na Seção 4.4 foi discutido um conjunto restrito de métricas.

Outra limitação está relacionada à estratégia de desenvolvimento do processo. Humphrey sugere que esta etapa seja fortemente baseada em observação. Alguns estudos de caso foram realizados, ou seja, foi solicitado aos estudantes que executassem as tarefas pré-determinadas. Não foi considerado se havia motivação para desempenhar a tarefa e como eles iriam proceder se pudessem escolher a forma de desempenhá-la.

Processo para Projetos de Pesquisa em Software

5.1 Considerações iniciais

Neste capítulo é apresentado o processo proposto para desenvolvimento de projetos de pesquisa em software. Após uma breve discussão, em que são retomados os principais elementos que foram importantes para a definição do processo, é apresentado o processo padrão. Em seguida são apresentados o processo especializado e um exemplo de instanciação da proposta. O modelo CMMI foi utilizado para apoiar a etapa de especialização.

A instanciação do processo apresentado nesta tese foi realizada no contexto do desenvolvimento do projeto SAFE (*Software Engineering Available for Everyone*). O projeto SAFE foi desenvolvido por membros do ICMC-USP, da UFMS e da empresa Async Open Source, sediada em São Carlos. O projeto teve início em 2004 e foi finalizado em 2006, sendo que seu principal objetivo foi o desenvolvimento de uma infra-estrutura que permitisse integrar ferramentas de software livre de apoio às atividades de Engenharia de Software. Buscou-se oferecer um suporte automatizado para o processo de software livre, que fosse simples o suficiente para atrair a colaboração e a participação de desenvolvedores, independente do seu nível de familiaridade com o processo de software livre.

Neste capítulo são discutidos também aspectos da avaliação do processo proposto.

5.2 Fundamentos para a elaboração do processo proposto

Para a definição do processo padrão foram considerados aspectos específicos do desenvolvimento de projetos distribuídos e colaborativos e do desenvolvimento de software livre. Além disso, conforme sugerido na literatura, buscou-se combinar elementos de métodos ágeis e tradicionais (Segal, 2005). É importante destacar também os processos que tornam viáveis o estabelecimento de parcerias entre universidades e empresas.

É possível notar, como características semelhantes entre o desenvolvimento de projetos de software livre e o desenvolvimento de projetos de pesquisa, a evolução contínua do projeto, em que novas versões são frequentemente disponibilizadas; a ausência de um conjunto determinado de requisitos no início do desenvolvimento do projeto; a dispersão geográfica dos membros e a distribuição gratuita do software desenvolvido em grande parte dos projetos. Em especial, as experiências e as práticas executadas pela comunidade de software livre contribuíram para a definição dos processos de coordenação, de comunicação e de gerenciamento.

O desenvolvimento distribuído e colaborativo é uma das principais características do desenvolvimento de projetos de pesquisa de interesse neste trabalho. Assim, em particular, práticas indicadas por Maidantchik (1999), na proposta de um processo para o desenvolvimento de projetos distribuídos, e por Humphrey (1999), na proposta de um processo para desenvolvimento de projetos por equipes, foram enfatizadas. De forma geral, conforme apresentado no Capítulo 2, processos importantes neste contexto se referem ao gerenciamento de configuração, comunicação, gerenciamento e coordenação de projetos, estabelecimento da infra-estrutura, gerenciamento de requisitos, planejamento e controle de projetos e garantia da qualidade.

Dois importantes valores defendidos nas propostas de métodos ágeis para desenvolvimento de software foram considerados no contexto deste trabalho. Em um dos valores é enfatizado que software funcional é mais importante que documentação detalhada. Foi observado, conforme apresentado no Capítulo 4, que em geral há pouca experiência com a execução de boas práticas para desenvolvimento de projetos de pesquisa. Como sugerido por Humphrey (1995), propor uma mudança drástica nos processos executados correntemente não é viável. Assim, considerando este valor, foi proposta uma documentação simplificada das atividades a serem executadas, porém, incluindo elementos essenciais.

Outro valor importante considerado na proposta de métodos ágeis está relacionado a priorizar a resposta a mudanças, ao invés do cumprimento de um plano. Trazendo para o contexto do desenvolvimento de pesquisas, nota-se que também há a prioridade em investigar respostas para as questões de pesquisa quando elas surgem. Assim, a obtenção

de conhecimento, a proposta de soluções inovadoras e as investigações para a resolução dos problemas de pesquisa também devem ser considerados como primordiais.

Conforme observado por Jenkins (2004) e Macke et al. (1996), é difícil implementar a parceria entre academia e indústria por diversos motivos, dentre eles, a falta de interesse de membros da academia em projetar sistemas para a área de negócios e em elaborar materiais apropriados para oferecer treinamento para a indústria. No entanto, apesar das dificuldades, membros da academia, da indústria e do governo têm buscado desenvolver projetos conjuntamente e, sobretudo, aprender com os erros, propor modelos de parceria, estender seus domínios e melhorar seus negócios e suas pesquisas de acordo com as experiências obtidas.

Diversos autores, baseados em suas experiências, indicaram atividades consideradas fundamentais para que a parceria entre universidade e indústria aconteça. Kornecki et al. (2003) destacaram a importância em definir claramente objetivos, papéis e responsabilidades, usar uma abordagem incremental para o desenvolvimento de software, elaborar documentação completa dos projetos, incluindo elementos de *design rationale*. Jenkins (2004) destaca a importância do gerenciamento efetivo do projeto. Macke et al. (1996) indicaram também a importância da definição clara de objetivos e papéis e destacaram a importância da formação de equipes cujos membros se comunicam com facilidade, da divisão do projeto em fases, do uso de um processo iterativo, da compatibilidade entre os treinamentos oferecidos e dos interesses das entidades envolvidas.

Em ambiente de desenvolvimento de pesquisas observa-se que os requisitos são identificados primordialmente de forma iterativa (Segal, 2005). Além disso, há interesse no desenvolvimento de protótipos, com o objetivo de validar resultados de pesquisa. Estes protótipos geralmente são continuados por diversos estudantes e pesquisadores e, conforme observado por Boldyreff et al. (2004), é fundamental produzir documentação que contribua para garantir a evolução das investigações e a obtenção de resultados científicos.

5.3 Processo padrão para desenvolvimento de projetos de pesquisa em software

Para a descrição do processo padrão, foi utilizada uma estrutura semelhante àquela apresentada na norma ISO/IEC 15504, parte 5. Portanto, conforme apresentado no Apêndice B, para cada processo são descritas suas atividades, sendo que para cada atividade são apresentados (a) identificador, (b) proposta, (c) resultados e (d) tarefas. Para cada tarefa são apresentados (a) objetivos, (b) artefatos de entrada e (c) artefatos de saída.

Na Figura 5.1 é apresentada uma visão geral do processo padrão para o desenvolvimento de projetos de pesquisa em software. Processos apresentados em caixas com fundo

branco são equivalentes a processos apresentados na norma ISO/IEC 12207 e processos apresentados em caixas com fundo cinza foram incluídos como resultado da realização deste trabalho.

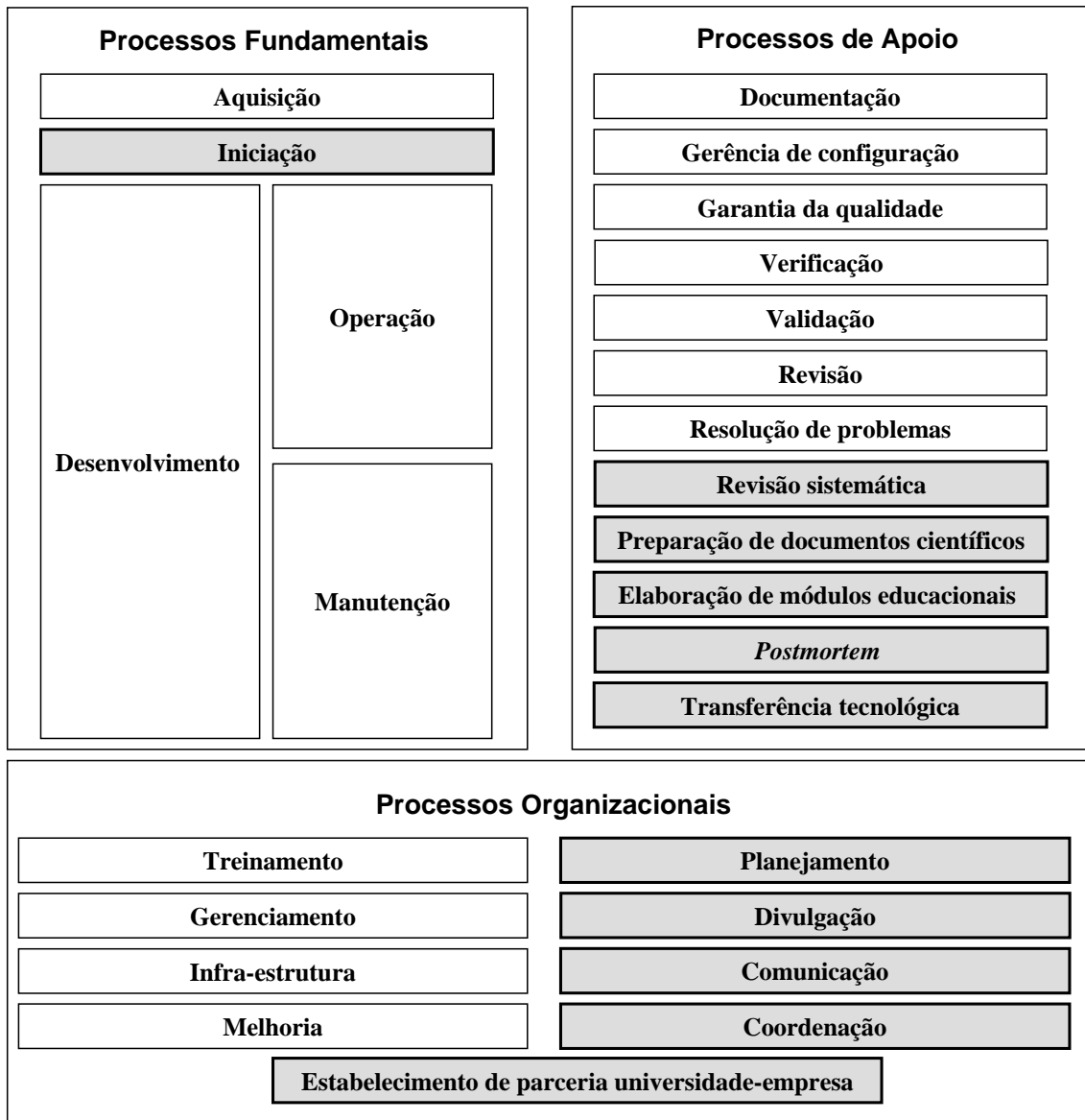


Figura 5.1: Processo padrão para o desenvolvimento de projetos de pesquisa em software

A descrição dos processos considerando-se o contexto de desenvolvimento de projetos de pesquisa é apresentada no Apêndice B. Conforme indicado no Capítulo 2, é fundamental apresentar *guidelines* que auxiliem a execução do processo, incluindo, por exemplo, atividades e artefatos a serem produzidos.

5.4 Especialização do processo padrão para projetos de pesquisa em software

De acordo com a proposta apresentada por Maidantchik (1999), o processo padrão deve ser especializado de forma a indicar uma aplicação específica para um determinado contexto. A principal motivação para a especialização do processo padrão é permitir que gerentes e coordenadores de equipes possam entender quais partes do processo padrão atendem aos diferentes níveis de um modelo de maturidade, por exemplo, o CMMI. Dessa forma, a especialização requer que seja estabelecida a correspondência entre os diferentes aspectos do processo padrão e as áreas apresentadas no modelo de maturidade. Assim, é possível identificar os aspectos do processo os quais devem ser enfatizados a fim de atender às áreas de um determinado nível de maturidade.

Neste trabalho, com o objetivo de atender às diferentes características das equipes que participam do desenvolvimento de projetos de pesquisa, foi realizada a especialização do processo padrão de forma a permitir que cada equipe utilize um processo com nível de maturidade compatível à sua realidade. Para especializar o processo foi estabelecido como referência o modelo CMMI (CMMI, 2006) (representação por estágios), no qual foram definidos os diferentes níveis de maturidade. Os resultados obtidos são apresentados na Tabela 5.1 (para especialização em relação ao nível 2 do CMMI) e na Tabela 5.2 (para especialização em relação ao nível 3 do CMMI). Foi observado que para os níveis 4 e 5 do CMMI não há processos correspondentes no processo padrão elaborado. Conforme apresentado no Capítulo 7, o estudo da viabilidade da inclusão dos processos referentes a estes níveis é um dos trabalhos futuros propostos.

5.5 Instanciação do processo padrão para projetos de pesquisa em software

Conforme apresentado por Maidantchik (1999), a atividade de instanciação consiste na seleção e alocação de métodos, técnicas de desenvolvimento, recursos humanos e tecnológicos para cada uma das atividades de um projeto particular. É interessante notar que há uma relação entre a atividade de instanciação proposta por Maidantchik e a atividade de validação do processo, proposta na abordagem de Humphrey (1995) para a definição de processos de software. É possível considerar que a atividade de instanciação contribui para a validação e os seus resultados são fundamentais para que sejam propostas melhorias nos processos padrão e especializado.

Nesta seção é apresentado um exemplo de instanciação do processo padrão para desenvolvimento de projetos de pesquisa em software. Serão apresentados elementos relacionados a recursos humanos, recursos computacionais, métodos de desenvolvimento

Tabela 5.1: Especialização do processo padrão para o nível 2 do CMMI. No Apêndice B encontram-se as descrições das atividades e tarefas que estão apresentadas nesta tabela por meio de seus identificadores.

Modelo CMMI (nível 2)	Processo Padrão para Projetos de Pesquisa em Software		
Áreas	Processos	Atividades	Tarefas
Gerenciamento de requisitos	Gerenciamento	GN	GN.1
Planejamento de projeto	Planejamento	DE, PI, PZ, PC, PS	DE.1, DE.2, PI.1, PI.2, PI.3, PZ.1, PC.1, PS.1, PS.2, PS.3
	Iniciação	ET, EP	ET.1, ET.2, ET.3, ET.4, EP.1, EP.2
	Documentação	PD	PD.1
	Infra-estrutura	IT, IF	IT.1, IT.2, IF.1
Monitoramento e controle de projeto	Coordenação	MR, EC	MR.1, EC.1
—	Aquisição	AQ, AR	AQ.1, AQ.2, AR.1, AR.2, AR.3
Gerência de subcontratação	Estabelecimento de parceria universidade-empresa	EA	EA.1, EA.2, EA.3
Medição e análise	—	—	—
Garantia de qualidade de processo e de produto	Garantia da qualidade	IC, GQ, GA	IC.1, GQ.1, GA.1
	Documentação	IP, PO	IP.1, PO.1
	Revisão	IB, RR	IB.1, RR.1, RR.2, RR.3
	Resolução de problemas	ID, RP	ID.1, RP.1
Gerenciamento de configuração	Gerência de configuração	IO, II, CC, AC	IO.1, II.1, CC.1, CC.2, CC.3, AC.1
—	Revisão sistemática	IM, RE	IM.1, RE.1, RE.2, RE.3
—	Preparação de documentos científicos	RD, FD	RD.1, FD.1
—	Elaboração de módulos educacionais	EM	EM.1, EM.2, EM.3
—	Divulgação	IA, DI	IA.1, DI.1, DI.2
—	Comunicação	IN, PC	IN.1, PC.1, PC.2

e recursos computacionais utilizados para o desenvolvimento do projeto SAFE (Fortes et al., 2004b).

Recursos Humanos

A equipe original do projeto SAFE, conforme descrita na proposta submetida em 2004, era composta por 32 pessoas, sendo 3 coordenadores (um de cada uma das entidades

Tabela 5.2: Especialização do processo padrão para o nível 3 do CMMI. No Apêndice B encontram-se as descrições das atividades e tarefas que estão apresentadas nesta tabela por meio de seus identificadores.

Modelo CMMI (nível 3)	Processo Padrão para Projetos de Pesquisa em Software		
	Processos	Atividades	Tarefas
Desenvolvimento de requisitos	Iniciação	AE	AE.1, AE.2
	Desenvolvimento	DF	DF.1
Solução técnica	Iniciação	PA	PA.1
	Desenvolvimento	PR, PP, DP, AP	PR.1, PP.1, DP.1, DP.2, DP.3, AP.1, AP.2, AP.3
	Operação	GU, SU	GU.1, SU.1
	Manutenção	PM, RM	PM.1, RM.1, RM.2, RM.3, RM.4
Integração do produto	Desenvolvimento	IR	IR.1, IR.2, IR.3
Verificação	Verificação	IE, VE	IE.1, VE.1
Validação	Validação	IS, VA	IS.1, VA.1
Foco no processo organizacional	Melhoria	DR, AO, MP	DR.1, AO.1, AO.2, MP.1, MP.2
Definição do processo da organização	–	–	–
Treinamento organizacional	Treinamento	IL, TR	IL.1, TR.1
Gerenciamento integrado de projeto	Gerenciamento	GE, GR, GC, GT, GO, GS	GE.1, GR.2, GR.3, GR.4, GR.5, GC.1, GC.2, GT.1, GO.1, GS.1
Gerenciamento de riscos	Gerenciamento	GI	GI.1
Análise de decisão e resolução	–	–	–
–	Postmortem	PE, FA	PE.1, PE.2, FA.1, FA.2
–	Transferência Tecnológica	PT, RS	PT.1, PT.2, RS.1, RS.2, RS.3

envolvidas: ICMC-USP, UFMS e Async Open Source); os demais membros assumiram os papéis de analistas e desenvolvedores.

Por ser realizado principalmente em ambiente acadêmico, onde existe alta rotatividade dos pesquisadores envolvidos, ao final do primeiro ano do desenvolvimento do projeto a equipe sofreu algumas alterações, principalmente devido ao fato de que alguns alunos de mestrado e de graduação finalizaram seus cursos. Assim, cinco membros foram desligados do projeto e, por outro lado, sete novos membros passaram a compor a equipe. Portanto, no segundo ano de desenvolvimento, 34 pessoas compuseram a equipe de desenvolvimento do projeto SAFE.

Os principais papéis associados aos membros das equipes e os processos pelos quais estiveram responsáveis foram:

1. **Coordenador geral:** responsável pelos processos de iniciação, aquisição, garantia da qualidade, planejamento, gerenciamento, estabelecimento de parceria universidade-empresa, comunicação, transferência tecnológica, divulgação, verificação, validação, preparação de documentos científicos e revisão;
2. **Coordenador local:** responsável pelos processos de coordenação, garantia da qualidade, gerenciamento, comunicação, divulgação, transferência tecnológica, verificação, validação, preparação de documentos científicos e revisão;
3. **Analista:** responsável pelos processos de desenvolvimento, operação, manutenção, preparação de documentos científicos, gerência de configuração e transferência tecnológica;
4. **Desenvolvedor:** responsável pelos processos de desenvolvimento, operação, manutenção, gerência de configuração, preparação de documentos científicos e transferência tecnológica;
5. **Instrutor:** responsável pelos processos de elaboração de módulos educacionais e treinamento.

Processos executados

Alguns processos que constituem o processo padrão para desenvolvimento de projetos de pesquisa em software foram executados no desenvolvimento do projeto SAFE. De forma a exemplificar, são descritas a seguir atividades e tarefas de desenvolvimento, planejamento e gerenciamento cumpridas. Outras informações sobre a execução dos processos no contexto do projeto SAFE (em especial as lições aprendidas) foram apresentadas em Paiva et al. (2006c).

- **Desenvolvimento**

No início do projeto foram realizadas diversas reuniões entre os membros para a compreensão do objetivo do projeto e a identificação dos requisitos iniciais. Os participantes foram motivados a expressar suas opiniões a respeito da proposta do projeto, identificar casos de uso do software e indicar possíveis direcionamentos para o desenvolvimento do projeto (uso de métodos, ferramentas, plataformas e arquiteturas). A maioria das reuniões foi realizada em formato de *brainstorming*, cuidando-se do planejamento e do registro (por meio de atas) dos itens apresentados e discutidos em cada reunião. Como resultado dessas reuniões foram gerados

diagramas de casos de uso, descrição dos casos de uso e modelo de dados por duas das instituições participantes (ICMC-USP e Async Open Source). Toda documentação do projeto foi publicada em uma *wiki* específica do projeto¹. A comunicação ocorreu principalmente por *emails*. Informações relativas ao planejamento e ao registro das informações das reuniões, questionamentos e soluções técnicas foram divulgadas a todos os membros.

Foi elaborado um projeto rápido do sistema e um protótipo foi desenvolvido pelos membros da terceira instituição participante (UFMS). Os resultados do desenvolvimento inicial do projeto foram discutidos no I Workshop SAFE, em abril de 2005, em que estiveram presentes representantes de todas as instituições envolvidas. Foram tomadas diversas decisões relacionadas ao planejamento das atividades do projeto, que também foram registradas em formato de ata, enviadas por *email* aos participantes e armazenadas na *wiki* do projeto.

A partir do desenvolvimento do protótipo, toda a equipe manifestou que foi possível entender melhor e refinar os requisitos do software. Como consequência, atualizações foram realizadas nos diagramas inicialmente gerados. Com esta evolução, decidiu-se por dividir o trabalho de implementação entre os membros de cada instituição envolvida, reforçando a abordagem colaborativa, tanto na tomada de decisões como na realização das tarefas de desenvolvimento propriamente. Os resultados obtidos foram então apresentados no II Workshop SAFE, em novembro de 2005. Novamente estiveram presentes membros das três instituições envolvidas. Com as discussões ocorridas, os casos de uso do sistema foram novamente refinados e a implementação continuou a evoluir. É importante observar que durante a realização dos *workshops*, em que as atividades executadas e os resultados obtidos foram apresentados, sentiu-se a necessidade da elaboração de outros diagramas (principalmente diagrama de classes e de arquitetura do sistema), de forma a facilitar o entendimento e a evolução do projeto. Considerando-se que o projeto foi desenvolvido de forma distribuída e colaborativa, comumente os membros das diferentes instituições participantes precisavam entender e dar continuidade a partes do sistema desenvolvido por outros membros. Um problema identificado neste sentido foi a falta de documentação de testes realizados, que gerava incertezas a respeito da qualidade da implementação.

Para o desenvolvimento do projeto SAFE não foi estabelecido inicialmente um modelo de ciclo de vida que deveria ser utilizado e, na prática, a abordagem iterativa foi adotada e evidenciada como sendo naturalmente necessária.

• Planejamento e Gerenciamento

¹Inicialmente a documentação estava disponível em <http://safe.icmc.usp.br/coteia>; atualmente está disponível em <http://safe.icmc.usp.br/safe>

No início do desenvolvimento do projeto foram geradas planilhas referentes ao planejamento inicial e que serviram como um artefato de entrada para o gerenciamento das atividades do projeto. Nas planilhas foram indicados relacionamentos entre as metas do projeto e as pessoas responsáveis; as metas e os prazos; e as metas, prazos e artefatos que deveriam ser elaborados.

As planilhas foram úteis na atribuição inicial das metas aos participantes e na definição de prazos e de artefatos a serem gerados. Observando-se que a atualização das planilhas estava se tornando uma tarefa pouco eficiente à medida que o volume de informações aumentava, optou-se por utilizar um sistema de gerenciamento *online*, o NetOffice². As pessoas envolvidas foram motivadas a publicar informações referentes ao desenvolvimento de suas atividades. Apesar de a equipe ter sido treinada para usar a ferramenta e a importância em utilizá-la ter sido enfatizada, poucas informações foram registradas. Grande parte das informações era registrada por um dos membros do projeto, de acordo com as atividades e os resultados apresentados nas reuniões.

De forma similar ao que acontece em ambiente industrial, notou-se que não houve entre os participantes a cultura de documentar, passo a passo, as atividades que executam. No geral, apenas os resultados finais de cada atividade foram apresentados na forma de um artefato, o que prejudicou o entendimento do projeto quando novos membros passaram a compor a equipe.

Recursos computacionais

A ferramenta OpenOffice Calc³ foi utilizada para a elaboração das planilhas de planejamento e gerenciamento. Outras ferramentas de software livre foram utilizadas para apoiar o desenvolvimento, a comunicação entre os membros e a documentação do projeto. A ferramenta PHPDoc⁴ foi utilizada para a documentação do código desenvolvido em PHP, as ferramentas CVS⁵ e Subversion⁶ foram utilizadas para controle de versões dos artefatos desenvolvidos, o CMS Plone⁷ foi utilizado para gerenciamento de conteúdo, a ferramenta Bugzilla⁸ para controle de alterações e a ferramenta NetOffice⁹ para gerenciamento do projeto.

A instanciação do processo padrão no desenvolvimento do projeto SAFE cobriu apenas partes dos processos, atividades e tarefas propostos. Considerando-se que o projeto

²<http://netoffice.sourceforge.net/>

³<http://www.openoffice.org/product/calc.html>

⁴<http://www.phpdoc.org/>

⁵<http://www.nongnu.org/cvs/>

⁶<http://subversion.tigris.org/>

⁷<http://plone.org/>

⁸<http://www.bugzilla.org/>

⁹<http://netoffice.sourceforge.net/>

SAFE foi desenvolvido de forma distribuída e colaborativa, as dificuldades relacionadas ao gerenciamento do projeto prejudicaram, muitas vezes, o entendimento sobre o estado corrente de desenvolvimento ou a percepção sobre o projeto. As dificuldades foram notadas, por exemplo, quando havia necessidade de identificar quais metas e qual a porcentagem de cada atividade havia sido executada. Além disso, um número limitado de ferramentas foi utilizado. Experiências interessantes poderiam ter sido obtidas caso plataformas e ferramentas tais como a plataforma GENESIS e a ferramenta MILOS (mencionadas no Capítulo 2) tivessem sido utilizadas.

É importante destacar também que a experiência com a instanciação do processo padrão sugeriu a inclusão de um novo processo, que se refere ao apoio à experimentação. De fato, esta é uma prática comum em ambiente de desenvolvimento de pesquisas e a investigação sobre sua inclusão pode ser sugerida como um trabalho futuro.

5.6 Avaliação do processo padrão para projetos de pesquisa em software

Foi realizada uma avaliação simplificada do processo, que visa a contribuir para a validação da proposta (última etapa da proposta de Humphrey (1995) para definição de processos de software). A avaliação consistiu em:

1. verificar quais requisitos, características e propriedades apresentados na literatura para definição de processos genéricos e discutidos no Capítulo 2 foram atendidos;
2. comparar a proposta com o estado da arte.

Os resultados dessa avaliação são apresentados nas sub-seções seguintes.

5.6.1 Em relação a requisitos, características e propriedades de processos

Inicialmente o processo proposto foi avaliado em relação às propriedades de processos identificadas por Pfleeger (2001), Madhavji (1991) e Kellner (1988) e apresentadas na Seção 2.2. A avaliação foi importante no sentido de indicar elementos fundamentais que precisam ser melhorados. As propriedades e a discussão no contexto do processo proposto são apresentadas a seguir:

- *O processo possui razões que suportam suas definições?* Sim. Foi observado na literatura a importância em cumprir processos para o desenvolvimento de projetos de pesquisa em software. As propostas existentes contemplavam parcialmente os requisitos identificados para um processo de desenvolvimento de projetos de pesquisa.

- *O processo possui as perspectivas funcionais, comportamentais e organizacionais?* Não. Somente a perspectiva funcional foi considerada.
- *O processo descreve e integra diferentes níveis de abstração?* Não. Foi apresentada apenas uma descrição textual do processo.
- *O processo apresenta sintaxe e semântica definidas formalmente?* Está sendo desenvolvido um projeto na UFMS com o objetivo de dar início à representação sintática do processo. O projeto tem como objetivo fazer um levantamento das linguagens de modelagem de processo (PML – *Process Modeling Language*), analisar suas vantagens e desvantagens e aplicar uma delas na modelagem do processo proposto. Um trabalho relacionado à formalização de processos de software foi apresentado por Reis (2003a). Pretende-se investigar a possibilidade de adoção dos mecanismos de especificação formal explorados por Reis, em especial a gramática de grafos, no contexto deste trabalho. Pretende-se propor também investigações sobre a aplicação de ontologias e linguagens formais tais como OWL (*Web Ontology Language*)¹⁰ e RDF (*Resource Description Framework*)¹¹ para a descrição semântica do processo proposto. Um trabalho relacionado à representação semântica de processos foi apresentado por Yang et al. (2006). A viabilidade de aplicação dos resultados obtidos pelos autores no contexto deste trabalho deverá ser analisada.
- *O processo é reutilizável?* Não foi possível avaliar, pois depende da utilização do processo na prática.
- *O processo é compreensível em abrangência e em detalhes?* Não foi possível avaliar, pois depende da utilização por usuários finais.

5.6.2 Em relação ao estado da arte

É importante destacar inicialmente que não foram indicadas nas propostas apresentadas na literatura as atividades executadas para a concepção dos processos. Assim, na maioria das propostas não foram especificados os objetivos e os requisitos para a definição dos processos e o que se espera obter com a execução dos mesmos. Acredita-se, no entanto, que estas informações sejam fundamentais para promover a evolução das propostas. Além disso, de forma geral, não foi observado nos trabalhos apresentados na literatura a motivação para a execução de atividades relacionadas à especificação do processo (representação sintática e semântica) e ao uso e à melhoria do processo. Portanto, atividades como modelagem do processo, implementação de um ambiente que ofereça suporte automatizado e validação do processo não foram mencionadas pelos autores.

¹⁰<http://www.w3.org/TR/2004/REC-owl-features-20040210/>

¹¹<http://www.w3.org/RDF/>

A proposta apresentada por Chirouze et al. (2005), denominada *eXtreme Researching*, estende o método XP para o contexto de desenvolvimento de pesquisas, considerando um ambiente de distribuição. No entanto, um das principais atividades que devem ser executadas para viabilizar o desenvolvimento distribuído é o gerenciamento e poucas práticas relativas ao gerenciamento foram enfatizadas pelos autores. O processo apresentado por Hwang e Park (2006) é semelhante à proposta apresentada nesta tese em termos da estrutura utilizada como base para definição da proposta (os autores utilizaram a norma ISO/IEC 15288 (2002), que trata de processos de ciclo de vida para desenvolvimento de sistemas). Porém, atividades fundamentais como o apoio ao desenvolvimento distribuído e colaborativo não foram enfatizados. No HDG (Walker, 2003), não há uma visão de processo, e sim uma proposta de práticas base para o desenvolvimento de projetos de pesquisa. É interessante notar, como um fator positivo, a simplicidade na apresentação da proposta, pois os autores exploraram o uso de termos comuns do desenvolvimento de pesquisas para descrevê-la. A proposta apresentada por Nunamaker e Chen (1990) enfatiza essencialmente atividades de ciclo de vida do desenvolvimento de um projeto.

Neste trabalho, buscou-se atender os requisitos apresentados na Seção 4.4 para um processo de desenvolvimento de projetos de pesquisa. É possível observar que a maioria dos requisitos foram cobertos pelas propostas apresentadas na literatura, porém, de forma individual. Portanto, o processo descrito nesta tese pode ser observado como uma proposta inicial de integração desses requisitos em uma única abordagem. As limitações da proposta são apresentadas na Seção 7.2.

5.7 Considerações finais

Neste capítulo foi apresentada a proposta de um processo para desenvolvimento de projetos de pesquisa em software, um dos principais resultados desta tese. Um conjunto de requisitos identificado na literatura e resultados de estudos de caso realizados fundamentaram a elaboração da proposta.

Em particular, as dificuldades em relação à documentação dos projetos de pesquisa, relatados por alguns autores na literatura, serviram como motivação para que uma alternativa para melhoria do processo de documentação fosse investigada. Os estudos realizados e os resultados obtidos são apresentados no próximo capítulo.

Um Modelo de *Design Rationale* para Projetos de Pesquisa em Software

6.1 Considerações iniciais

Neste capítulo é apresentado o estudo realizado sobre a aplicação de *design rationale* no processo de desenvolvimento de projetos de pesquisa em software, tendo em vista a melhoria da documentação e enfatizando especificamente a atividade de análise de requisitos de software.

6.2 Experiências com *design rationale* em ambiente acadêmico

Nos últimos quatro anos, foram realizados estudos de caso envolvendo a captura e o registro de *design rationale* no ICMC-USP, em disciplinas dos cursos de graduação e pós-graduação em Ciência da Computação. Durante o desenvolvimento dos projetos foi solicitado aos estudantes que indicassem suas opiniões a respeito do uso da abordagem, principalmente em relação ao auxílio à manutenção dos projetos. Conforme observado por Burge (2005), uma dificuldade em investigar o potencial uso de *design rationale* no desenvolvimento de software é que há poucas bases de dados relacionadas para análise. Assim, a atividade inicial proposta foi criar uma base que auxiliasse na realização das investigações.

Foram realizados estudos de caso em 6 disciplinas e 124 alunos estiveram envolvidos. A perspectiva da argumentação foi explorada, sendo que os modelos IBIS e PHI e as

ferramentas DocRationale (Francisco, 2004) e DocRat (Lara et al., 2005) foram utilizados (nos primeiros estudos realizados não haviam ferramentas disponíveis e as informações foram registradas utilizando um editor de textos). De forma geral, foi solicitado aos estudantes que capturassem e registrassem informações de *design rationale* para os projetos desenvolvidos em grupos.

Após a criação de uma base de *design rationale* inicial, buscou-se avaliar se os estudantes estavam de fato registrando as informações relacionadas às discussões e decisões durante a etapa de desenvolvimento. Certamente, as informações de *design rationale* não poderão ser úteis na fase de manutenção se não forem convenientemente tratadas na fase de desenvolvimento. Assim, uma das atividades do trabalho de Panosso (2004) foi analisar a base obtida, contando a quantidade de questões, posições e argumentos que foi registrada nas quatro fases desenvolvidas (análise de requisitos, projeto, prototipação rápida e testes) em 31 projetos avaliados das 6 disciplinas. Na Tabela 6.1 são apresentados os valores médios obtidos que representam o volume de dados registrado para cada item de argumentação. Por exemplo, considerando a primeira linha e a segunda coluna da tabela é possível observar que em média 2,41 questões foram registradas na fase de análise de requisitos em cada projeto.

Tabela 6.1: Informações de *design rationale* registradas em 31 projetos acadêmicos de software (valores médios)

	Questões	Posições	Argumentos
Análise de requisitos	2,41	1,16	0,67
Projeto	4,61	1,32	0,73
Prototipação rápida	1,8	0,91	0,84
Testes	0,94	1,17	0,25

É notável que poucas informações de *design rationale* foram de fato registradas pelos estudantes. Na prática, observou-se que os estudantes tinham dificuldades em organizar as informações de acordo com as estruturas de argumentação propostas nos modelos IBIS e PHI. As observações realizadas no decorrer do desenvolvimento dos projetos, permitiram identificar dois possíveis fatores que justificam o pouco interesse dos estudantes para o registro de *design rationale*:

- Os estudantes não são engenheiros de software experientes e a atividade de documentação não é familiar a eles. Assim, produzir uma rede elaborada de argumentação, categorizando e ligando informações convenientemente não foi uma atividade simples para desenvolvedores novatos. Foi observado que, em muitas situações, havia dificuldade em discutir argumentos favoráveis e contra, provavelmente por não possuírem experiência prática suficiente para discutir diferentes alternativas para os problemas que ocorreram.

- Nos modelos tradicionais para *design rationale*, alguns nós e *links*, por exemplo, questões e argumentos, não refletem o processo de Engenharia de Software, o que se torna um empecilho para a adoção dos modelos.

6.3 Modelo de representação de *design rationale* para desenvolvimento de projetos de pesquisa em software

Como resultado das experiências realizadas, foi elaborado um conjunto de requisitos para um modelo de representação de *design rationale* (Paiva et al., 2006a), conforme apresentado a seguir.

1. *Foco no objetivo de cada fase do processo*: a captura de diferentes tipos de informação, de acordo com o objetivo de cada fase do processo deve ser possível. Portanto, nas fases de análise de requisitos e de projeto, o foco está em “o que” constituirá o software e “porque”. Por outro lado, durante a fase de implementação, *design rationale* pode ser útil para registrar informações relacionadas a “como” os requisitos foram codificados. Acredita-se que a especialização direcione a captura de informações úteis minimizando os problemas relacionados ao gerenciamento de uma vasta quantidade de informações. Além disso, é possível que o uso de *design rationale* seja aprimorado porque os analistas e programadores poderão distinguir quais tipos de informações poderão ser recuperadas em cada fase do processo de desenvolvimento do software.
2. *Foco nas decisões, justificativas e resultados das experiências – fases de análise de requisitos e de projeto*: a captura de decisões tomadas, justificativas e resultados das experiências deve ser possível. A captura de argumentos a favor e contra, sugerida em muitos modelos tradicionais, não é um elemento primordial. Considera-se que projetos de pesquisa sejam desenvolvidos em ambiente de aprendizado e realização de experiências, em que as alternativas estão sendo avaliadas e experimentadas.
3. *Rastreabilidade*: a associação entre os requisitos do software e as informações de *design rationale* deve ser possível. A associação permite identificar posteriormente, em fases mais adiantadas do processo, quais decisões foram tomadas em relação a cada requisito, ou seja, recuperar informações adicionais sobre eles.
4. *Inclusão de comentários*: a inclusão de opiniões, lições aprendidas e sugestões para novas investigações deve ser possível. Em ambiente de desenvolvimento de pesquisas, o raciocínio empregado para resolver um problema, as reflexões, a experimentação e as discussões são elementos fundamentais para a obtenção dos resultados científicos e podem resultar na obtenção de conhecimentos valiosos para as equipes. A

inclusão de comentários oferece, portanto, a possibilidade de registrar esse tipo de informação, utilizando-se um formato livre.

5. *Evolução de design rationale*: a evolução das informações de *design rationale* deve acompanhar a evolução das versões do software, ou seja, deve ser possível associar as informações de *design rationale* às versões dos artefatos. Em termos de desenvolvimento de software, a atividade de manutenção é muito intensiva e, portanto, torna-se importante distinguir claramente quais decisões foram tomadas durante o desenvolvimento de cada versão do projeto. O relacionamento entre *design rationale* e versões pode trazer muitos benefícios. Por exemplo, membros novatos podem compreender, passo a passo, como a evolução do software ocorreu e, além disso, uma documentação indicando quais decisões são válidas para cada versão é gerada, reforçando o requisito de rastreabilidade. Este requisito não foi coberto na proposta apresentada nesta tese. No entanto, um projeto de pesquisa, que deverá ser implementado como um projeto de mestrado, foi proposto recentemente com o objetivo de investigar mecanismos para implementação deste requisito no contexto da proposta.
6. *Informações de contexto*: informações relacionadas ao contexto no qual o *design rationale* foi elaborado devem ser capturadas. Isso inclui identificar quem registrou determinada informação de *design rationale*, quando (data/hora) e onde (em qual projeto/em qual versão) ela foi registrada. Este requisito também não foi coberto na proposta apresentada. Pretende-se propor um projeto de iniciação científica cujo objetivo será a implementação deste requisito.

Outro requisito importante considerado neste trabalho está relacionado à simplicidade do modelo de *design rationale*. Como apresentado por Souza et al. (1999), as informações resultantes das discussões muitas vezes são perdidas devido à fraca motivação das equipes em registrá-las. De acordo com Regli et al. (2000), um desafio é reduzir o custo para registrar *design rationale* e garantir que sua organização permita a recuperação posterior. Neste sentido, observando-se que um dos focos dos métodos ágeis é a simplicidade da documentação, foi verificado como os princípios defendidos no Manifesto Ágil¹ poderiam ser aplicados no contexto de *design rationale*.

Conforme apresentado a seguir, foi identificada a possibilidade de aplicar cinco dos princípios, aconselhados no Manifesto Ágil, no contexto deste trabalho. O objetivo foi definir uma estrutura simples para registro das informações de *design rationale* e motivar os membros de um projeto de pesquisa para seu uso. Assim, os princípios seguidos foram os seguintes:

¹<http://agilemanifesto.org/>

1. *Entrega preliminar e contínua do software*: este princípio sugere que o software seja desenvolvido iterativamente e que os resultados obtidos sejam avaliados pelos usuários. Em termos de *design rationale* isso significa capturar informações valiosas desde fases preliminares do processo de software até a manutenção. Além disso, as informações de *design rationale* devem poder ser capturadas de forma iterativa e colaborativa.
2. *Desenvolver projetos com pessoas motivadas (oferecer o ambiente e o suporte necessários)*: este princípio sugere que sejam fornecidas as condições necessárias para que as pessoas de fato se envolvam com o projeto. Em termos de *design rationale* isso significa fornecer informações que contribuam para o desenvolvimento, o entendimento e a manutenção dos projetos. Além disso, deve ser possível utilizar as informações como uma base para discutir e raciocinar sobre um projeto desenvolvido de forma colaborativa.
3. *Software funcional é a principal medida de progresso*: este princípio sugere que o software funcionando é mais importante que a documentação completa e detalhada. Em termos de *design rationale* isso significa que as informações de *design rationale* podem ser valiosas e contribuir para o progresso do projeto, somente se forem acessadas para auxiliar projetistas a tomar decisões ou resolver algum problema durante o desenvolvimento e a manutenção (Burge e Brown, 2000). Há interesse, portanto, em compreender em quais fases do processo de software *design rationale* pode ser útil e qual a estrutura de representação deve ser utilizada em cada uma delas.
4. *Simplicidade*: este princípio sugere que a solução mais simples possível deve sempre ser desenvolvida. Em termos de *design rationale* isso significa manter a estrutura para registro das informações simples, de forma a minimizar os custos de uma documentação adicional.
5. *Soluções diretas*: este princípio sugere que soluções simples e fáceis de entender sejam desenvolvidas. Em termos de *design rationale* isso significa que o modelo de representação utilizado seja fácil de compreender e seja bem documentado.

Considerando-se os requisitos apresentados, foi elaborado um modelo de representação de *design rationale* com o objetivo de atender o contexto de desenvolvimento de pesquisas em software. O foco é o registro de *design rationale* nas fases de análise de requisitos e projeto. De forma geral, os requisitos do software representam um elemento de entrada, para os quais as informações de *design rationale* são registradas.

O modelo de representação, denominado **DR-SACI** (**D**esign **R**ationale of **S**oftw**A**re developed in resear**Ch** env**I**ronment), é apresentado na Figura 6.1. Em seguida seus principais elementos conceituais são descritos.

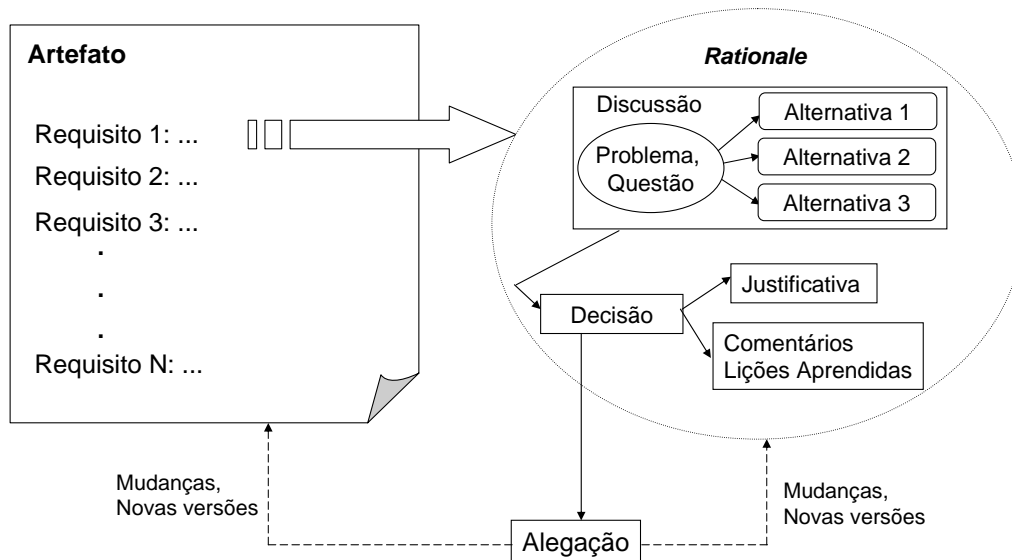


Figura 6.1: Modelo de representação de *design rationale* para o contexto de desenvolvimento de projetos de pesquisa em software (DR-SACI)

- *Artefato*: representa o artefato de software que contém os requisitos elicitados;
- *Requisito*: representa um requisito funcional ou não funcional do sistema. Dúvidas ou problemas relacionados à elicitação dos requisitos levam a discussão ou reflexão. Como resultado, tal discussão entre os participantes do desenvolvimento de um projeto pode melhorar o entendimento e o detalhamento dos requisitos;
- *Problema, Questão*: representam os questionamentos ou as dúvidas que surgem na fase de elicitação dos requisitos do projeto. Está relacionado a qualquer problema ou dificuldade da equipe que leve a uma discussão;
- *Alternativas*: representam possíveis soluções para as questões ou dúvidas;
- *Decisão*: representa a indicação da alternativa selecionada. Pode estar relacionada a uma combinação de alternativas;
- *Justificativa*: representa uma explicação sobre “porque” determinada decisão foi tomada;
- *Comentários, Lições aprendidas*: representam os registros sobre os resultados obtidos a partir das decisões que foram tomadas, sobre trabalhos futuros (por exemplo, propor uma modificação em determinado requisito para realizar novas investigações) e para detalhar as lições aprendidas;
- *Alegação*: representam informações que indicam a necessidade de reflexão em relação a requisitos especificados anteriormente ou decisões tomadas, tendo em vista a realização de novas experiências e a evolução do projeto de pesquisa. Por

exemplo, problemas ou desvantagens de uma decisão que foi tomada ou sugestão para selecionar outra alternativa para o mesmo problema podem ser indicadas;

O relacionamento entre os elementos conceituais do modelo **DR-SACI** foi definido de forma que as setas em **traço contínuo** indicam relacionamento de associação, mostrando uma relação entre dois conceitos, no sentido de complementar a informação que se tem sobre eles em um determinado instante. Por exemplo, uma decisão tomada está associada a uma justificativa.

No modelo **DR-SACI**, as setas em **traço pontilhado** simbolizam o relacionamento de evolução de um projeto de pesquisa, ou seja, a obtenção de novas versões, tanto de um artefato que contém os requisitos, quanto da documentação de *design rationale*, geradas como resultado de uma alegação a uma decisão.

6.4 Ferramenta para apoiar o registro de *design rationale* de projetos de pesquisa em software

A partir da proposta do modelo **DR-SACI** para representação de *design rationale*, foi realizada uma adaptação na ferramenta MVCASE² de forma a implementar parcialmente os requisitos e o modelo apresentados na Seção 6.3 (Paiva et al., 2006b).

A ferramenta MVCASE está sendo desenvolvida por alunos de graduação e pós-graduação da Universidade Federal de São Carlos (UFSCar) e do Instituto de Ciências Matemáticas e de Computação (ICMC-USP). Desenvolvida em Java, é uma ferramenta livre que teve suas origens como uma ferramenta de modelagem (Prado e Lucrédio, 2000), e está evoluindo desde então, com funcionalidades sendo acrescentadas à sua arquitetura, como desenvolvimento baseado em componentes (Almeida et al., 2002; Prado e Lucrédio, 2001), integração com ambientes de Engenharia de Software (Lucrédio et al., 2004), desenvolvimento de software orientado a aspectos (Garcia et al., 2004) e suporte à reutilização de software (Lucrédio, 2005).

A ferramenta MVCASE é extensível por meio de *plug-ins*. Assim, os desenvolvedores podem adicionar funcionalidades extras às funções básicas de modelagem, sem a necessidade de modificar seu código-base. Um desenvolvedor pode incluir funcionalidades em *plug-ins* que podem ser instalados facilmente na ferramenta. Um dos *plug-ins* desenvolvidos para a MVCASE trata a representação de *design rationale*.

Na Figura 6.2 é apresentada uma tela da ferramenta MVCASE indicando a implementação do modelo de representação de *design rationale*. Pode ser observado na figura que um elemento correspondente a um item de *design rationale* foi disponibilizado na paleta de elementos gráficos (indicado por (1)). Utilizando o botão, é possível incluir, em qualquer diagrama suportado pela ferramenta, textos referentes à documentação segundo

²<http://mvcase.dev.java.net>

o modelo apresentado na Figura 6.1. O texto incluído fica então visível em uma caixa (indicado por (2)), dividida em quatro compartimentos: *Problema*, *questão e alternativa* (representados por “P”); *Decisões* (representadas por “D”); *Justificativas* (representados por “J”) e *Comentários e Lições Aprendidas* (representado por “Comments”). É possível associar um item de *design rationale* diretamente a um determinado elemento do diagrama, por meio de uma linha pontilhada.

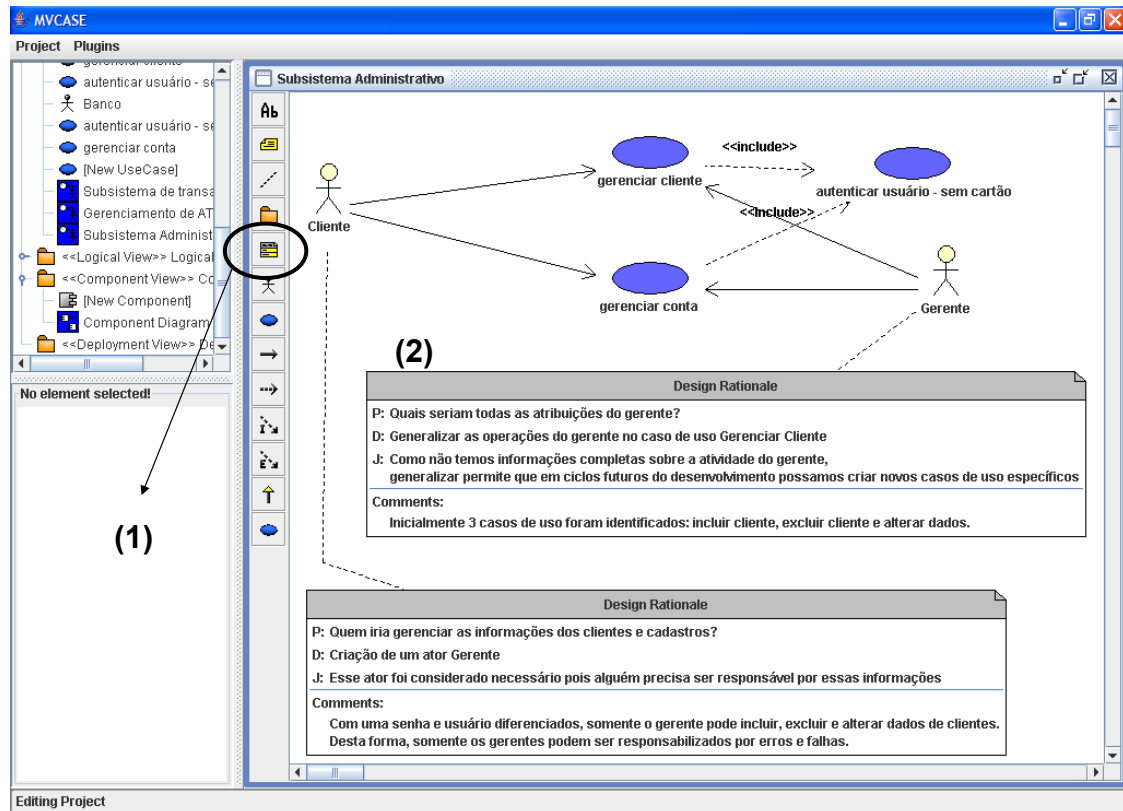


Figura 6.2: Exemplo de uso da ferramenta MVCASE em que informações de *design rationale* foram capturadas

Conforme observado por Regli et al. (2000), diversos fatores contribuem para que informações de *design rationale* não sejam capturadas na prática durante o desenvolvimento de projetos. Com a realização deste trabalho (elaboração do modelo proposto e implementação na MVCASE), buscou-se diminuir os efeitos desses fatores e motivar os desenvolvedores para a captura das informações. Um dos fatores está relacionado ao fato de geralmente se propor a utilização de uma nova ferramenta para captura de *design rationale*, o que gera uma sobrecarga adicional ao processo de software. Na ferramenta MVCASE, é possível incluir as informações nos próprios diagramas, sem a necessidade de utilizar outra ferramenta. Outro fator apresentado por Regli et al. é que os sistemas de *design rationale* não suprem as necessidades específicas de determinados contextos, por exemplo, uma empresa ou um determinado ambiente. No caso da ferramenta

MVCASE, foi implementado um modelo diferenciado, em que foi considerado o domínio de desenvolvimento de pesquisas em software.

6.5 Experimento envolvendo o modelo de *design rationale* e a ferramenta MVCASE

Foi realizado um experimento em duas disciplinas do curso de Ciência da Computação do ICMC-USP. Os modelos IBIS e DR-SACI foram utilizados, constituindo o tratamento 1 e o tratamento 2, respectivamente. Foi solicitado aos estudantes que capturassem e registrassem *design rationale* para os projetos que foram propostos, que deveriam ser desenvolvidos em grupo e de forma colaborativa.

O experimento consistiu da análise de três diferentes projetos de software, com um nível similar de dificuldade, considerando-se o número de requisitos e o número de classes do software.

O enunciado do primeiro projeto estava relacionado à distribuição de disciplinas para professores de uma universidade (sistema 1, desenvolvido por estudantes da disciplina 1, usando o modelo IBIS); o do segundo projeto estava relacionado ao controle de bilhetes eletrônico para transporte municipal (sistema 2, desenvolvido por estudantes da disciplina 2, usando o modelo DR-SACI); e o do terceiro projeto estava relacionado ao controle de caixa eletrônico de um banco (sistema 3, desenvolvido por estudantes da disciplina 2, usando o modelo DR-SACI). As disciplinas 1 e 2 foram respectivamente: “Sistemas de Informação” e “Análise e Projeto Orientados a Objetos”.

Nas sub-seções seguintes são apresentadas as atividades executadas para a definição, o planejamento e a execução de experimentos, de acordo com os *guidelines* recomendados por Wohlin et al. (2000), envolvendo o modelo DR-SACI e a ferramenta MVCASE (Paiva et al., 2007).

6.5.1 Definição do experimento

Objeto de estudo: modelo para registro de *design rationale* para o contexto de desenvolvimento de projetos de pesquisa em software (DR-SACI).

Proposta: avaliar o modelo DR-SACI em relação à motivação para registro de *design rationale* e sua utilidade na manutenção e evolução de software desenvolvidos em projetos de pesquisa.

Foco qualitativo: eficácia do modelo DR-SACI.

Perspectiva: o experimento foi realizado sob o ponto de vista de estudantes do curso de graduação do ICMC-USP.

Contexto: foram apresentadas diferentes especificações de requisitos aos estudantes e foi solicitado que cumprissem atividades de desenvolvimento e manutenção de software. Foi oferecido treinamento (aproximadamente 3 horas) sobre conceitos de *design rationale* e sobre os modelos e as ferramentas utilizadas. Estudantes da disciplina 1 cursavam o oitavo período do curso e os estudantes da disciplina 2 cursavam o sétimo período. O estudo envolvendo o modelo IBIS foi realizado no primeiro semestre de 2005 e o estudo envolvendo o modelo DR-SACI foi realizado no primeiro semestre de 2006.

6.5.2 Planejamento do experimento

Seleção do contexto: o experimento foi realizado por sessenta estudantes divididos em vinte grupos (três estudantes em cada grupo). O experimento foi realizado fora da sala de aula e sem supervisão e, portanto, os estudantes tiveram a liberdade de determinar seus próprios cronogramas para cumprir as atividades propostas.

Hipótese nula (H0): a quantidade de itens de *design rationale* registrados usando o modelo DR-SACI é menor ou o mesmo que a quantidade de itens de *design rationale* registrados usando um modelo tradicional.

Hipótese alternativa (H1): a quantidade de itens de *design rationale* registrados usando o modelo DR-SACI é maior que a quantidade de itens de *design rationale* registrados usando um modelo tradicional.

Projeto do experimento: o experimento foi projetado para ser executado em duas fases. Na Fase 1, o foco estava na modelagem dos três sistemas, usando a linguagem UML. Os estudantes das duas disciplinas cumpriram esta fase. Na Fase 2, foi solicitado aos estudantes que tentassem compreender os projetos desenvolvidos por outros grupos para que pudessem realizar uma atividade de manutenção que foi proposta. Apenas os estudantes da disciplina 2 participaram desta segunda fase. Os grupos que desenvolveram o sistema 2 realizaram a manutenção no sistema 3 e vice-versa.

Na Fase 1, o objetivo foi fazer uma análise quantitativa dos resultados, ou seja, para cada um dos sistemas foi contado o número de itens de *design rationale* registrados pelos estudantes na fase de análise de requisitos. No modelo DR-SACI foi considerado como um item de *design rationale*, um conjunto de informações contendo problema - alternativa(s) - decisão - justificativa (comentários são opcionais). Para o modelo IBIS, cada conjunto

de informações de questão - posições - argumentos foi contado como um item de *design rationale*.

Na Fase 2, o objetivo foi fazer uma análise qualitativa dos resultados, ou seja, avaliar a importância de *design rationale* na evolução e manutenção do software. Aproximadamente metade dos grupos pôde acessar e avaliar a importância das informações de *design rationale* produzidas na Fase 1. Os grupos que não utilizaram as informações de *design rationale* produzidas na Fase 1 puderam avaliar o quanto as informações poderiam facilitar o entendimento dos projetos durante a realização da manutenção.

Na Tabela 6.2 são apresentadas informações relacionadas às duas fases do experimento, em relação aos grupos que estiveram responsáveis por cumpri-las e aos sistemas que foram utilizados (Legenda: DDP = sistema para Distribuição de Disciplinas para Professores; CBE = sistema para Controle de Bilhetes Eletrônicos para transporte municipal; CCE = sistema para Controle de Caixa Eletrônico de um banco; '-' = estudantes não participaram).

Tabela 6.2: Projeto do experimento

Disciplina	Identificador do grupo	Fase 1: Análise requisitos	Fase 2: Manutenção	Uso de <i>design rationale</i>
1	1	DDP	-	-
1	2	DDP	-	-
1	3	DDP	-	-
1	4	DDP	-	-
1	5	DDP	-	-
1	6	DDP	-	-
1	7	DDP	-	-
1	8	DDP	-	-
1	9	DDP	-	-
2	10	CBE	CCE	<i>sim</i>
2	11	CCE	CBE	<i>sim</i>
2	12	CBE	CCE	não
2	13	CCE	CBE	não
2	14	CBE	CCE	<i>sim</i>
2	15	CCE	CBE	<i>sim</i>
2	16	CBE	CCE	não
2	17	CCE	CBE	não
2	18	CBE	CCE	<i>sim</i>
2	19	CCE	CBE	<i>sim</i>
2	20	CBE	CCE	não

Variáveis independentes

Metodologia usada – na Fase 1 (modelagem do software), aproximadamente metade dos estudantes usou o modelo IBIS e os demais usaram o modelo DR-SACI. Na Fase 2 (manutenção do software), apenas o modelo DR-SACI foi usado, realizando-se a troca de projetos entre os grupos. Os itens de *design rationale* elaborados na Fase 1 foram disponibilizados a aproximadamente metade dos estudantes durante a Fase 2.

Experiência dos estudantes e área de interesse: os estudantes possuíam experiência prévia em análise orientada a objetos e interesse em diferentes áreas da Ciência da Computação.

Variáveis dependentes

Quantidade de itens de design rationale registrados pelos participantes – Conforme mencionado anteriormente, foi definido previamente como os itens seriam contados. Em relação à Fase 2, as informações subjetivas dos participantes considerando a utilidade de *design rationale* na manutenção de software foi considerada uma variável dependente.

Seleção dos participantes

Os participantes foram escolhidos considerando-se a conveniência. A divisão dos estudantes em grupos não foi aleatória, pois eles puderam fazer as escolhas dos membros de acordo com suas vontades. A distribuição dos sistemas entre os grupos foi aleatória na Fase 1 e foi invertida na Fase 2. Os estudantes não tiveram a liberdade de escolher entre participar ou não do estudo, pois o experimento constituiu parte de um projeto obrigatório que deveria ser desenvolvido nas disciplinas. Portanto, não é possível garantir que os resultados seriam os mesmos se os participantes fossem voluntários.

Ameaças à validade

Validade de conclusão: para todos os grupos, os itens de *design rationale* foram contados considerando o critério mencionado no item *Projeto do experimento*. É importante observar, no entanto, que os estudantes receberam pontos de crédito nas disciplinas por suas participações no estudo. Há o risco de que o incentivo tenha influenciado os resultados apresentados. Além disso, o experimento foi realizado em duas disciplinas diferentes. Se os grupos são muito heterogêneos, é possível que as variações que ocorreram devido às influências individuais tenham interferido no resultado geral do experimento.

Validade interna

Os tratamentos não foram aplicados a todos os participantes, ou seja, é possível que um resultado diferente tivesse sido obtido se os tratamentos fossem trocados entre os grupos. Em relação a ameaça à maturação, a maioria dos participantes aparentemente estava concentrada em suas atividades (não houve controle em relação a esse efeito). Na Fase 2, em relação à instrumentação, é importante notar que os resultados são dependentes do conteúdo dos itens de *design rationale* gerados na Fase 1. Portanto, os resultados sobre a utilidade de *design rationale* para manutenção de software são influenciados pelas condições da base fornecida, por exemplo, completude. Se as informações não foram registradas adequadamente, os resultados foram afetados negativamente.

Validade de construção

O treinamento em *design rationale* não foi extensivo e, portanto, os conceitos e as definições provavelmente não foram claras para alguns estudantes. Poucos tipos de medições foram executadas e as conclusões podem ter sido afetadas negativamente. O fato de que os estudantes estavam desenvolvendo um projeto que é parte de uma disciplina pode ser considerado uma ameaça, pois eles poderiam tentar manipular as informações para melhorar suas notas. No entanto, esse fato não foi considerado, pois os estudantes foram informados a respeito da importância da acurácia das informações. Além disso, o professor da disciplina 1 não foi o mesmo da disciplina 2, o que pode ter causado diferenças no direcionamento da proposta das atividades.

Validade externa

Não é possível generalizar os resultados para diferentes contextos pois, para a elaboração do modelo considerado no experimento (DR-SACI) foram enfatizadas características comuns do contexto de desenvolvimento de projetos de pesquisa. Considerou-se aceitável a realização dos experimentos com estudantes sobre um modelo voltado para o desenvolvimento de projetos de pesquisa, pois sabe-se que em geral os estudantes são membros das equipes que desenvolvem os projetos.

Instrumentação

O material fornecido aos participantes para a execução dos experimento foi composto por: *guidelines* para executar o experimento, documento de requisitos do sistema (apenas o necessário para cada grupo), documentos contendo descrições do modelo IBIS e do modelo DR-SACI (apenas o necessário para cada grupo), documentos descrevendo diagrama de casos de uso e diagrama de classes, planilhas para coletar os dados do experimento.

6.5.3 Realização do experimento

Preparação

Como parte do treinamento, foram ministradas aulas sobre *design rationale* (conceitos, modelos, etc) e análise orientada a objetos. Os artefatos indicados no item *Instrumentação* foram disponibilizados na *web*.

Participantes

Os estudantes possuíam conhecimento prévio em Engenharia de Software, mas eles não possuíam conhecimento sobre conceitos de *design rationale*.

Realização do experimento

Inicialmente, a descrição do sistema com o qual o trabalho seria realizado foi apresentado aos estudantes, bem como as tarefas que deveriam ser executadas (modelagem do sistema e registro de *design rationale*). Foi considerado que avaliar a motivação dos estudantes em termos da quantidade de itens de *design rationale* capturados não era suficiente, ou seja, seria fundamental obter alguma informação sobre a qualidade das informações em relação à utilidade na fase de manutenção dos projetos. Dessa forma, foi proposta uma atividade de manutenção para estudantes da disciplina 2, usando o modelo DR-SACI. Os estudantes da disciplina 1 não participaram porque o tema de manutenção de software não fazia parte do programa do curso. Os projetos desenvolvidos por estudantes da disciplina 2 foram trocados entre os grupos e um novo requisito foi adicionado. Foi solicitado que as modificações necessárias fossem realizadas. Os dados resultantes das duas fases foram coletados da seguinte forma: após a Fase 1, a documentação disponibilizada pelos estudantes das duas disciplinas foi examinada e o número de itens de *design rationale* foi contado. Durante a execução da Fase 2, os grupos registraram em um questionário as suas opiniões sobre a importância das informações de *design rationale* durante a manutenção de software. As respostas obtidas também foram contabilizadas.

Validação dos dados

Os questionários distribuídos aos participantes foram verificados com o objetivo de garantir que foram preenchidos corretamente. Os registros obtidos foram satisfatórios e nenhum questionário foi descartado.

6.5.4 Resultados dos experimentos

Primeiramente, as hipóteses serão analisadas, considerando-se os dados quantitativos. Em seguida, as respostas obtidas para o questionário serão discutidas.

Os resultados da análise estatística descritiva considerando-se as duas disciplinas são apresentados na Tabela 6.3. A terceira coluna apresenta os valores médios relacionados à quantidade de itens de *design rationale* registrados pelos estudantes usando ambos os modelos. A diferença entre os valores fornece uma evidência (embora não prove) de que a motivação para registro de *design rationale* melhorou quando o modelo DR-SACI foi utilizado.

Tabela 6.3: Resultados quantitativos em relação à captura de *design rationale*

	Quantidade de grupos	Itens de <i>design rationale</i> (valor médio)	Desvio padrão
Modelo IBIS (disciplina 1)	9	3,78	3,49
Modelo DR-SACI (disciplina 2)	11	8,36	2,42

Um gráfico *box plot* foi gerado para facilitar a visualização dos dados (Figura 6.3). A partir da figura é possível observar que há um padrão que indica que a quantidade de itens de *design rationale* registrada foi maior quando o modelo DR-SACI foi utilizado.

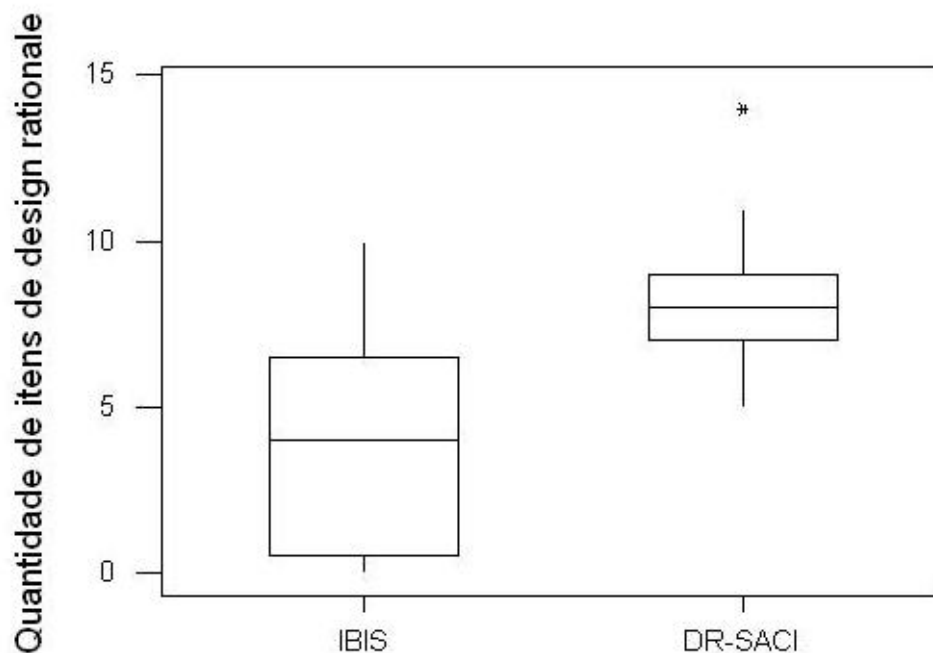


Figura 6.3: Quantidade de itens de *design rationale* registrados em relação ao uso dos modelos IBIS e DR-SACI

Os dados coletados foram testados em relação à normalidade por meio do teste Anderson-Darling, utilizando-se a ferramenta MINITAB. Os testes de hipótese para normalidade dos dados provenientes do uso do modelo IBIS ($p\text{-value}=0,470$) e dos dados provenientes do uso do modelo DR-SACI ($p\text{-value}=0,069$) não foram rejeitados, considerando o nível de significância de 5%. Os gráficos gerados são apresentados nas Figuras 6.4 e 6.5.

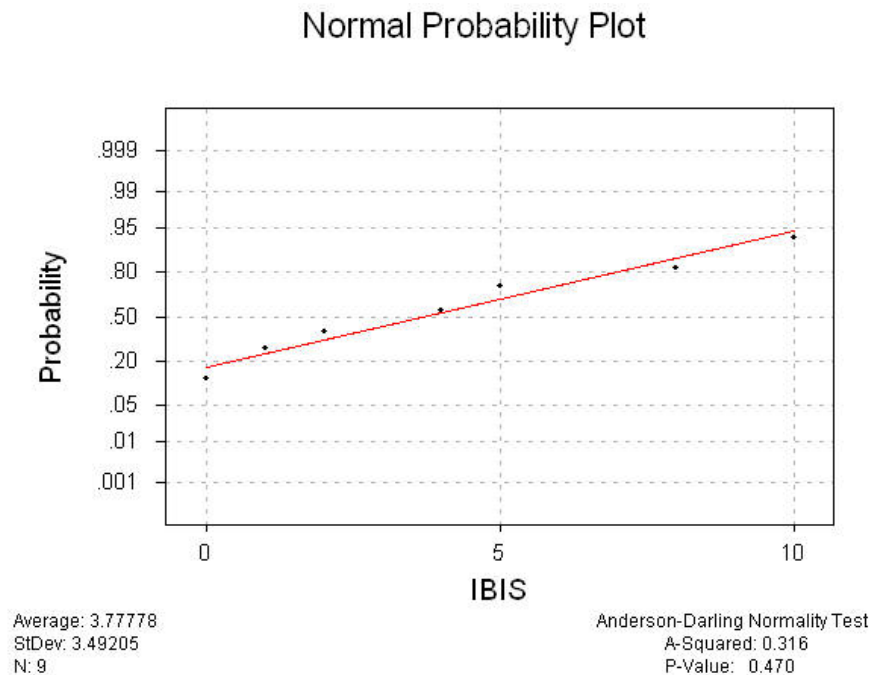


Figura 6.4: Resultados do teste de normalidade dos dados obtidos com o uso do modelo IBIS

Foi realizado um t-test bilateral com as duas amostras presumindo variâncias equivalentes. O resultado indicou que o número de itens de *design rationale* registrado para grupos que usaram o modelo DR-SACI foi significativamente maior em comparação ao uso do modelo IBIS ($p\text{-value}=0,0027$). Com este resultado, portanto, é possível rejeitar a hipótese nula (H_0) e concluir que há uma diferença significativa entre os resultados obtidos quando o modelo IBIS e DR-SACI foram utilizados.

Foi solicitado, ainda, que os estudantes da disciplina 2 registrassem suas opiniões durante o experimento sobre alguns tópicos específicos. Foi perguntado aos grupos que usaram *design rationale* durante a fase de manutenção se as informações apresentadas (em relação às categorias do modelo DR-SACI) foram suficientes para auxiliar a compreensão do projeto. Como apresentado na Figura 6.6, a maioria dos estudantes indicou que as categorias do modelo foram satisfatórias.

Os grupos que usaram *design rationale* durante a fase de manutenção também avaliaram se as informações disponibilizadas ajudaram a compreender o projeto desenvolvido

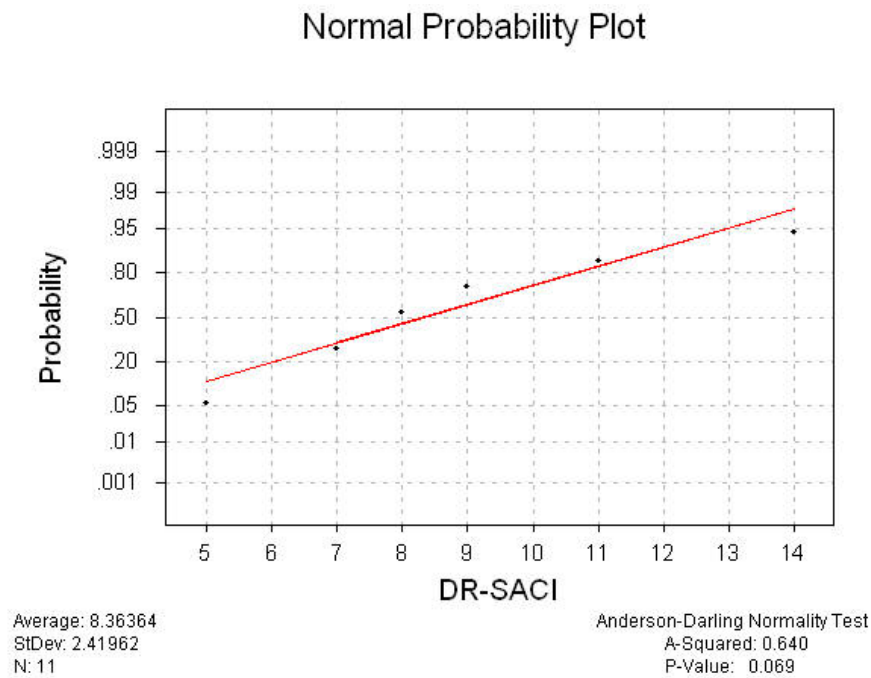


Figura 6.5: Resultados do teste de normalidade dos dados obtidos com o uso do modelo DR-SACI

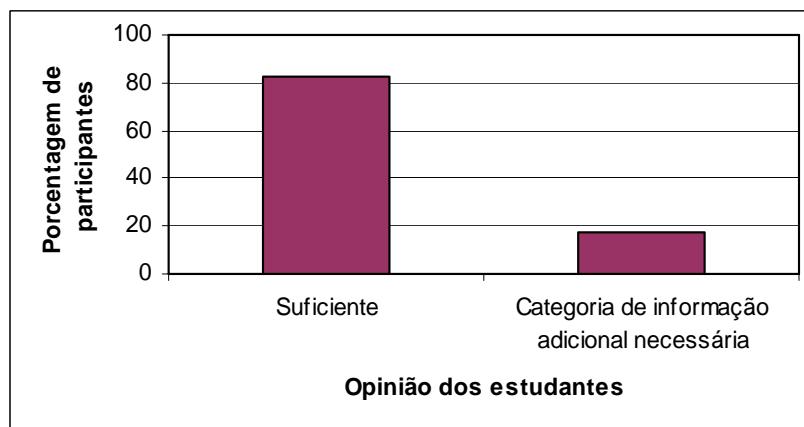


Figura 6.6: Resultados da avaliação da estrutura do modelo DR-SACI

por outro grupo. Como apresentado na Figura 6.7, aproximadamente 80% dos estudantes indicou que as informações de *design rationale* foram importantes ou muito importantes.

Foi perguntado aos grupos que realizaram a manutenção sem *design rationale* se, ao surgirem dúvidas durante a etapa de compreensão do projeto, era possível encontrar as respostas nos diagramas disponibilizados ou se informações adicionais teriam sido relevantes. Conforme apresentado na Figura 6.8, 60% dos participantes julgaram importante a disponibilização das informações adicionais. Por outro lado, 40% dos estudantes indicou que as informações adicionais não poderiam ajudá-los, porque as dúvidas surgiram principalmente em relação ao novo requisito considerado.

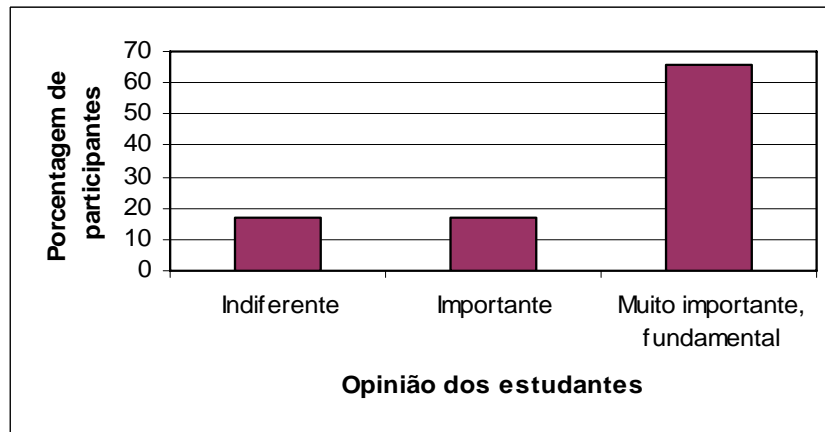


Figura 6.7: Resultados da avaliação da importância do modelo DR-SACI na fase de manutenção

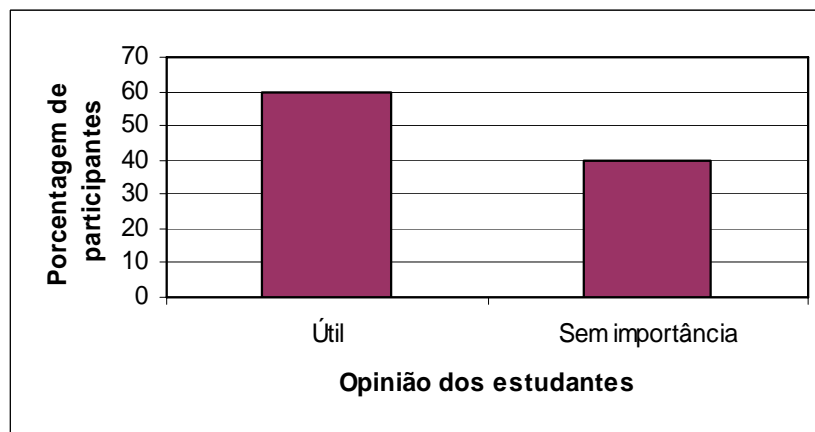


Figura 6.8: Opinião dos estudantes que realizaram a manutenção sem o uso de *design rationale*

Finalmente, é importante destacar outro resultado positivo relacionado ao experimento. Durante a fase de manutenção, o registro de *design rationale* não foi solicitado aos estudantes. No entanto, aproximadamente 65% deles fez o registro de tais informações. Este resultado foi observado como uma forte evidência de que, uma vez que os estudantes tenham compreendido a importância de *design rationale* em documentação de software, eles estarão motivados a usar a abordagem em projetos futuros.

6.6 Comparação entre propostas para registro de *design rationale* em projetos de pesquisa

Conforme apresentado na Seção 3.5, a única proposta encontrada na literatura envolvendo *design rationale* no contexto de projetos de pesquisa foi apresentada por Kim et al. (1993). Os autores propuseram uma adaptação do modelo IBIS para o contexto

de desenvolvimento de pesquisas científicas colaborativas. No entanto, as atividades executadas para elaboração do modelo e os resultados experimentais de sua avaliação não foram indicados.

Considerando-se a importância que a Engenharia de Software experimental tem assumido nos últimos anos (Wohlin et al., 2000), diferentemente da metodologia adotada por Kim et al, na proposta do modelo DR-SACI, a experimentação está sendo considerada uma importante atividade para auxiliar a avaliação e a melhoria da proposta.

O principal elemento que diferencia a proposta apresentada nesta tese é a adoção da abordagem de registrar principalmente “o que o pesquisador fez”. Assim, o registro de alternativas que não foram implementadas e de argumentos para os quais não há avaliação não são consideradas como foco da proposta. O objetivo principal é indicar as decisões e as justificativas de forma explícita, colaborando para o entendimento dos projetos na fase de manutenção.

6.7 Considerações finais

Neste capítulo foi apresentado o modelo de *design rationale* elaborado para o desenvolvimento de projetos de pesquisa. Foram descritas as atividades executadas para obtenção do modelo e os resultados parciais obtidos (por exemplo, a proposta dos requisitos e a adaptação realizada na ferramenta MVCASE).

A necessidade de realização de novos estudos de caso torna-se evidente. Em particular, observa-se a importância em realizá-los no próprio ambiente de desenvolvimento de pesquisas. A experiência obtida com a realização do estudo descrito na Seção 6.5 será importante na definição, no planejamento e na realização de outros experimentos.

Conclusões

7.1 Contribuições

Este trabalho teve por objetivo identificar um processo para desenvolvimento de projetos de pesquisa envolvendo software, tendo em vista o auxílio à evolução desses projetos. Uma das principais dificuldades relatadas na literatura está relacionada à carência de documentação em um ambiente em que há alta rotatividade dos membros que participam das equipes. Esta foi uma importante motivação para que o processo de documentação fosse investigado de forma mais detalhada, considerando-se a abordagem de *design rationale*.

As principais contribuições deste trabalho e as experiências relacionadas podem ser resumidas conforme apresentadas a seguir:

1. Identificação de um processo para desenvolvimento de projetos de pesquisa, utilizando-se uma abordagem sistemática proposta na literatura. O uso da abordagem proposta por Humphrey para a definição do processo foi relevante, pois indicou etapas fundamentais, como a identificação de requisitos para o processo, que guiaram efetivamente o desenvolvimento da proposta. Além disso, acredita-se que a documentação gerada com a execução de cada etapa contribuirá para a evolução e melhoria da proposta.
2. Identificação de um modelo de *design rationale* para projetos de pesquisa, considerando-se características do domínio. Estudos de caso foram realizados no decorrer do trabalho e os resultados obtidos, as experiências realizadas e as lições aprendidas fundamentaram a elaboração da proposta. Além disso, o fato de ter sido cumprido um ciclo de desenvolvimento para a obtenção do modelo,

incluindo identificação de requisitos, projeto, implementação e experimentação, foi fundamental no sentido de possibilitar o refinamento gradual da proposta.

Nota-se que as contribuições estão inseridas em um contexto de pesquisa que está sendo valorizado atualmente. Há diversos incentivos para que projetos de pesquisa sejam desenvolvidos de forma colaborativa, envolvendo pesquisadores de diferentes áreas de conhecimento, tendo em vista aos avanços científicos e tecnológicos. Como resultado, a comunidade científica está começando a propor discussões de amplo alcance em torno do tema de **processos** para o desenvolvimento de projetos de pesquisa. Um exemplo disso é a realização do *First International Workshop on Improving Research Productivity in Computing*¹, a ser realizado em agosto de 2007 na França, que tem por objetivo discutir *como* os projetos de pesquisa estão sendo desenvolvidos e quais são as alternativas para melhorar os processos utilizados.

7.2 Limitações

As principais limitações deste trabalho foram:

- Alguns requisitos identificados pelos usuários, por exemplo, gerenciamento de conhecimentos, não foram cobertos pois optou-se por propor uma solução inicial que deverá ser avaliada, melhorada e estendida de forma incremental.
- A descrição das atividades, tarefas e artefatos do processo é simplificada e oferece apenas uma visão geral do trabalho que precisa ser realizado. Uma descrição mais completa envolve uma sugestão sobre “como” cada tarefa pode ser executada e a indicação de pré-condições para a realização de cada tarefa.
- A abordagem iterativa, mencionada como um requisito de um processo para desenvolvimento de projetos de pesquisa em software, não foi experimentada. Para experimentá-la, é fundamental aplicar o processo em projetos reais e coletar dados sobre a execução do processo. Com isso, será possível sugerir alternativas para a execução do processo de forma iterativa, de acordo com a experiência obtida.
- A avaliação e a validação do processo proposto foram pouco enfatizadas, sendo necessária, principalmente, a realização de experimentos reais no contexto de desenvolvimento de projetos de pesquisa colaborativos e distribuídos.
- A representação relacionada a como implantar e usar o processo apresentado não foi apresentada. Uma possibilidade é a proposta de uma ordem de execução das atividades, considerando uma abordagem iterativa.

¹<http://www.iaria.org/conferences2007/IRP.html>

- A proposta envolvendo *design rationale* não foi analisada em relação às atividades de captura e recuperação das informações. É importante averiguar o quanto essas atividades podem ser executadas satisfatoriamente quando o modelo é utilizado.

7.3 Trabalhos futuros

Além das propostas de trabalhos futuros indicadas nos capítulos anteriores, podem ser destacadas:

- Definir formalmente a sintaxe e a semântica do processo.
- Desenvolver um ambiente que auxilie a execução das atividades propostas e o armazenamento e a recuperação de informações relacionadas aos projetos de pesquisa. Uma possibilidade é a investigação da adequação de ambientes que já foram desenvolvidos, por exemplo, a estação TABA (Rocha et al., 2005) nesse escopo. Certamente este será um facilitador para o cumprimento das etapas 7 e 8 propostas por Humphrey, que se referem à validação e melhoria do processo.
- Investigar se os modelos de avaliação de processos de software atuais são aplicáveis ao contexto de desenvolvimento de projetos de pesquisa. Caso não sejam, investigar quais elementos devem compor tais modelos. Observa-se que alguns pesquisadores estão começando a se interessar em avaliar os processos nos laboratórios de pesquisa, com o objetivo de atrair e viabilizar novas parcerias com outros centros de pesquisa e com o setor industrial. Um modelo de avaliação, portanto, complementa a proposta apresentada neste trabalho.
- Investigar as principais características que definem um ambiente *collaboratory* e determinar o quanto a proposta apresentada está alinhada aos seus objetivos, propondo e realizando as modificações necessárias. Um ponto em comum observado é o foco no desenvolvimento distribuído e, portanto, é viável analisar o quanto as propostas se complementam.
- Dar continuidade às investigações relacionadas a um modelo de maturidade para o desenvolvimento de projetos de pesquisa envolvendo software, observando a viabilidade em tratar a especialização dos níveis 4 e 5 do CMMI para esse contexto.
- Realizar experimentos com o modelo *DR-SACI* em um ambiente de desenvolvimento de pesquisas.
- Investigar a possibilidade de aplicação da abordagem de *design rationale* como uma forma de capturar o conhecimento dos pesquisadores em relação a cada um dos processos apresentados na Figura 5.1.

- Cobrir os requisitos de *evolução de design rationale* e *informações de contexto* do modelo de representação, discutidos na Seção 6.3. Uma possibilidade é propor a extensão da ferramenta MVCASE para atender a estes requisitos. Outra possibilidade é considerá-los no contexto do ambiente para execução das atividades relacionadas a projetos de pesquisa (mencionado no item 2).

7.4 Produção científica

A seguir estão as referências das publicações decorrentes dos estudos realizados e dos resultados obtidos durante o desenvolvimento da pesquisa descrita nesta tese.

Artigos completos em periódicos

1. Paiva, D. M. B.; Freire, A. P.; Lucrédio, D.; Braga, R. T. V.; Fortes, R. P. M. “Reinforcing Design Rationale in Software Projects Developed in Academic Environment.” *International Transactions on Systems Science and Applications*, 2007. Aceito para publicação.
2. Freire, A. P.; Paiva, D. M. B.; Fortes, R. P. M. “Implantação de Gestão Descentralizada de Recursos Acadêmicos - um estudo de caso.” *Revista Brasileira de Informática na Educação*, v. 12, n. 2, p. 78-85, 2004.

Artigos completos em congressos

1. Paiva, D. M. B.; Turine, M. A. S.; Fortes, R. P. M. “Processo de Software Livre em Ambiente Acadêmico: Experiências e Lições Aprendidas.” *Proceedings*. In: CLEI 2006 - XXXII Conferencia Latinoamericana de Informática, Chile, 2006.
2. Paiva, D. M. B.; Lucrédio, D.; Fortes, R. P. M. “MVCASE - Incluindo Design Rationale para Auxílio à Modelagem em Projetos de Pesquisa.” *Anais*. In: XVIII Sessão de Ferramentas do XX Simpósio Brasileiro de Engenharia de Software, Florianópolis-SC, p. 73-78, 2006.
3. Paiva, D. M. B.; Fortes, R. P. M. “Design Rationale in Software Engineering: A Case Study.” *Proceedings*. In: Seventeenth International Conference on Software Engineering & Knowledge Engineering (SEKE), Taiwan, p. 342-348, 2005.
4. Paiva, D. M. B.; Manduca, A. M.; Silva, M. A. G.; Quemello, L. J.; Braga, R. T. V.; Fortes, R. P. M. “Reforçando a Comunicação com Uso de uma Ferramenta de Software Livre em Ensino de Engenharia de Software.” *Anais*. In: XIII Workshop sobre Educação em Computação (WEI 2005), XXV Congresso da Sociedade Brasileira de Computação (SBC 2005), São Leopoldo-RS, 2005.

5. Paiva, D. M. B.; Fortes, R. P. M. “Model for Academic Software Development Including Design Rationale Elements.” *Proceedings*. In: I IFIP Academy on the State of Software Theory and Practice - PhD Colloquium, Porto Alegre-RS, 2005.
6. Cagnin, M. I.; Paiva, D. M. B.; Maldonado, J. C.; Penteado, R. D.; Fortes, R. P. M.; Germano, R. S. “From Design Rationale to Reengineering Rationale: Lessons Learned in a Maintenance Pilot Case Study.” *Proceedings*. In: 4 Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, Espanha, p. 231-243, 2004.
7. Paiva, D. M. B.; Freire, A. P.; Sanches, R.; Fortes, R. P. M. “Definindo, Implantando e Melhorando Processos de Software em Ambiente Acadêmico.” *Anais*. In: VI Simpósio Internacional de Melhoria de Processo de Software (SIMPROS), São Paulo-SP, p. 71-82, 2004.
8. Paiva, D. M. B.; Sanches, R.; Fortes, R. P. M. “Melhoria de Processo de Software no Contexto do Desenvolvimento Acadêmico de uma Agenda na *Web*.” *Anais*. In: 14o Congresso Internacional de Tecnologia de Software (CITS 2003), Curitiba-PR, p. 56-68, 2003.
9. Francisco, S. D.; Izeki, C. A.; Paiva, D. M. B.; Fortes, R. P. M. “Um Sistema de Apoio à Utilização de Design Rationale de Artefatos de Software.” *Proceedings*. In: CLEI 2003 - XXIX Conferencia Latinoamericana de Informática, Bolívia, 2003.
10. Francisco, S. D.; Izeki, C. A.; Paiva, D. M. B.; Fortes, R. P. M. “DocRationale: Um Auxílio a Design Rationale na Documentação de Software.” *Anais*. In: X Sessão de Ferramentas, XVII Simpósio Brasileiro de Engenharia de Software (SBES2003), Manaus-AM, p. 61-66, 2003.

Resumos

1. Paiva, D. M. B.; Freire, A. P.; Fortes, R. P. M. “Design Rationale in Academic Software Development: Requirements for a Representation Model.” *Proceedings*. In: Eighteenth International Conference on Software Engineering & Knowledge Engineering (SEKE), USA, p. 469-472, 2006.
2. Lara, S. M.; Freire, A. P.; Paiva, D. M. B.; Fortes, R. P. M. “Registering Design Rationale in Collaborative Software Projects: a Case Study.” *Proceedings*. In: 4th International Information and Telecommunication Technologies Symposium (I2TS), Florianópolis-SC, p. 163-164, 2005.
3. Freire, A. P.; Paiva, D. M. B.; Fortes, R. P. M. “Manutenção de Aplicações *Web* Utilizando Separação de Interesses: Um Estudo de Caso.” *Proceedings*. In: 4th

International Information and Telecommunication Technologies Symposium (I2TS), Florianópolis-SC, 2005.

4. Fortes, R. P. M.; Freire, A. P.; Vieira, V. H.; Paiva, D. M. B. “An Academic Web-Based Agenda and its Engineering Process.” *Proceedings*. In: VII Workshop Iberoamericano de Ingeniería de Requisitos Y Desarrollo de Ambientes de Software, Peru, p. 151-156, 2004.

Relatório técnico

1. Paiva, D. M. B.; Freire, A. P.; Fortes, R. P. M. “Web Engineering Process - a Case Study from Academic Development.” Relatório Técnico do ICMC-USP, n. 79, 2004.

Outras publicações

1. Freire, A. P.; Paiva, D. M. B.; Turine, M. A. S.; Fortes, R. P. M. “Using Screen Readers to Reinforce Web Accessibility Education.” *Proceedings*. To Appear in: 12th Annual Conference on Innovation and Technology in Computer Science Education, Escócia, 2007.
2. Martins Netto, O. A.; Paiva, D. M. B.; Pimentel, M. G. C. “Abordagem para Derivação de Regras de Usabilidade Especializadas em Contextos de Aplicação Específicos.” *Proceedings*. In: CLEI 2004 - XXX Conferencia Latinoamericana de Informática, Peru, p. 789-800, 2004.
3. Fortes, R. P. M.; Silva, E. A.; Paiva, D. M. B. “Utilizando a Norma ISO/IEC 14598-5 na Avaliação da Qualidade de Hiperdocumentos *Web*.” *Anais*. In: I Simpósio Brasileiro de Qualidade de Software (SBQS), Gramado, p. 128-142, 2002.

Referências Bibliográficas

- ABRAHAMSSON, P.; SALO, O.; RONKAINEN, J.; WARSTA, J. *Agile Software Development Methods – Review and Analysis*. VTT Publications 478, 2002.
- ACUNA, S. T.; JURISTO, N. Assigning People to Roles in Software Projects. *Software – Practice and Experience*, v. 34, n. 7, p. 675 – 696, 2004.
- ALMEIDA, E. S.; LUCRÉDIO, D.; BIANCHINI, C. P.; PRADO, A. F.; TREVELIN, L. C. Ferramenta MVCASE - Uma Ferramenta Integradora de Tecnologias para o Desenvolvimento Baseado em Componentes Distribuídos. In: *Sessão de Ferramentas do XVI Simpósio Brasileiro de Engenharia de Software (SBES 2002)*, Gramado - RS, 2002, p. 432–437.
- ALPERN, B.; AUGART, S.; BLACKBURN, S. M.; BUTRICO, M.; COCCHI, A.; CHENG, P.; DOLBY, J.; FINK, S.; GROVE, D.; HIND, M.; MCKINLEY, K. S.; MERGEN, M.; MOSS, J. E. B.; NGO, T.; SARKAR, V.; TRAPP, M. The Jikes Research Virtual Machine Project: Building an Open-Source Research Community. *IBM Systems Journal*, v. 44, n. 2, p. 399–417, 2005.
- AMARAL, D. C. *Arquitetura para Gerenciamento de Conhecimentos explícitos sobre Processo de Desenvolvimento de Produto*. Tese de Doutorado, Escola de Engenharia de São Carlos, Universidade de São Paulo, 215 páginas, 2001.
- AMBATI, V.; KISHORE, S. P. How can Academic Software Research and Open Source Software Development Help Each Other? In: *4th Workshop on Open Source Software Engineering, 26th International Conference on Software Engineering*, Scotland, Reino Unido, 2004, p. 5–8.
- ARAÚJO, R. M.; BORGES, M. R. S. The Role of Collaborative Support to Promote Participation and Commitment in Software Development Teams. *Software Process: Improvement and Practice*, p. 229–246, 2007.
- ARNOLD, J. E. Toward Collaborative Software Processes. In: *Ninth International Software Process Workshop*, Airlie, Estados Unidos: IEEE Computer Society, 1994, p. 107–109.

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS NBR 14724: Informação e Documentação – Trabalhos acadêmicos – Apresentação. 2002a.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS NBR 6023: Informação e Documentação – Referências – Elaboração. 2002b.
- ATHEY, S.; PLOTNICKI, J. An Evaluation of Research Productivity in Academic IT. *Communications of the Association for Information Systems*, v. 3, p. 1–20, 2000.
- AUER, K.; MILLER, R. *Extreme Programming Applied*. Addison-Wesley, 2002.
- AUGUSTIN, L.; BRESSLER, D.; SMITH, G. Accelerating Software Development Through Collaboration. In: *24th International Conference on Software Engineering*, Orlando, Estados Unidos, 2002, p. 559–563.
- AVERSANO, L.; LUCIA, A. D.; GAETA, M.; RITROVATO, P.; STEFANUCCI, S.; VILLANI, M. L. Managing Coordination and Cooperation in Distributed Software Processes: the GENESIS Environment. *Software Process: Improvement and Practice*, v. 9, n. 4, p. 239–263, 2004.
- BABAR, M. A.; TANG, A.; GORTON, I.; HAN, J. Industrial Perspective on the Usefulness of Design Rationale for Software Maintenance: a Survey. In: *QSIC '06: Proceedings of the Sixth International Conference on Quality Software*, Beijing, China: IEEE Computer Society, 2006, p. 201–208.
- BALABKO, P.; WEGMANN, A.; RUPPEN, A.; CLEMENT, N. Capturing Design Rationale with Functional Decomposition of Roles in Business Processes Modeling. *Software Process: Improvement and Practice*, v. 10, n. 4, p. 379–392, 2005.
- BALL, L.; LAMBELL, N.; ORMEROD, T.; SLAVIN, S.; MARIANI, J. Representing Design Rationale to Support Innovative Design Reuse: A Minimalist Approach. In: *Proceedings of the 4th Annual Design Research Thinking Symposium - MIT*, 1999.
- BARBOSA, E. F. *Uma Contribuição ao Processo de Desenvolvimento e Modelagem de Módulos Educacionais*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 253 páginas, 2004.
- BARROS, A. J. P.; LEHFELD, N. A. S. *Projeto de Pesquisa: Propostas Metodológicas*. 16 ed. Vozes, 1990.
- BASS, L.; CLEMENTS, P.; NORD, R. L.; STAFFORD, J. Capturing and Using Rationale for a Software Architecture. In: *A. H. Dutoit and R. McCall and I. Mistrík and B. Paech (eds.), Rationale Management in Software Engineering*, Springer, p. 255–272, 2006.

- BASTOS, L. R.; PAIXÃO, L.; FERNANDES, L. M.; DELUIZ, N. *Manual para a Elaboração de Projetos e Relatórios de Pesquisas, Teses, Dissertações e Monografias*. 6 ed. LTC, 2004.
- BATTIN, R. D.; CROCKER, R.; KREIDLER, J.; SUBRAMANIAN, K. Leveraging Resources in Global Software Development. *IEEE Software*, v. 18, n. 2, p. 70–77, 1994.
- BECK, K. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 1999.
- BECK, K.; BEEDLE, M.; BENNEKUM, A.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. Manifesto for Agile Software Development. 2001.
Disponível em <http://agilemanifesto.org/> (Último Acesso em 29/08/2006)
- BECKMAN, K.; KHAJENOORI, S. Collaborations: Closing the Industry-Academia Gap. *IEEE Software*, v. 14, n. 16, p. 49–57, 1997.
- BELLOTTI, V.; BURTON, R.; DUCHENEAUT, N.; HOWARD, M.; NEUWIRTH, C.; SMITH, I. XP in a Research Lab: The Hunt for Strategic Value. In: *Proceedings of the Third International Conference on eXtreme Programming (XP2002)*, Alghero, Italy, 2002, p. 56–61.
- BERNAT, A.; TELLER, P. J.; GATES, A.; DELGADO, N. Structuring the Student Research Experience. In: *Proceedings of the 5th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finlândia, 2000, p. 17–20.
- BEZROUKOV, N. Open Source Software Development as a Special Type of Academic Research. *First Monday – peer-reviewed journal on the internet*, v. 4, n. 10, 1999.
- BIRK, A.; DINGSOYR, T.; STALHANE, T. Postmortem: Never Leave a Project without it. *IEEE Software*, v. 19, n. 3, 2002.
- BLEEK, W.-G.; LILIENTHAL, C.; SCHMOLITZKY, A. Transferring Experience from Software Engineering Training in Industry to Mass University Education: The Big Picture. In: *CSEET '05: Proceedings of the 18th Conference on Software Engineering Education & Training*, Ottawa, Canadá: IEEE Computer Society, 2005, p. 195–203.
- BOLDYREFF, C.; NUTTER, D.; RANK, S. Communication and Conflict Issues in Collaborative Software Research Projects. In: *4th Workshop on Open Source Software Engineering, 26th International Conference on Software Engineering*, Scotland, Reino Unido, 2004, p. 14–17.

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML – Guia do Usuário*. 2 ed. Campus, 2006.
- BOWEN, S.; MAURER, F. Process Support and Knowledge Management for Virtual Teams Doing Agile Software Development. In: *Workshop on Cooperative Supports for Distributed Software Engineering Processes – 26th IEEE Annual International Computer Software and Application Conference*, Oxford, Inglaterra, 2002, p. 1118–1120.
- BOY, G. Supportability-Based Design Rationale. In: *Proceedings of the 6th IFCA Symposium on Analysis, Design and Evaluation of Man-Machine Systems*, Boston, Estados Unidos, 1995.
- BREITLING, H.; KORNSTADT, A.; SAUER, J. Design Rationale in Exemplary Business Process Modeling. In: *Rationale Management in Software Engineering*, Springer, p. 191–208, 2006.
- BRITO, R.; FERREIRA, P.; SILVA, K.; BURÉGIO, V.; LEITE, I. Uma Experiência na Implantação de Processo em uma Fábrica de Software Livre. In: *VI Simpósio Internacional de Melhoria de Processos de Software*, São Paulo - SP, 2004, p. 119–130.
- BURGE, J.; BROWN, D. C. Reasoning With Design Rationale. In: *Artificial Intelligence Design*, Kluwer Academic Publishers, 2000, p. 611–629.
- BURGE, J.; BROWN, D. C. *Discovering a Research Agenda for Using Design Rationale in Software Maintenance*. Computer Science Technical Report WPI-CS-TR-02-03, Worcester Polytechnic University, 2002.
- BURGE, J. E. *Software Engineering Using Design Rationale*. Tese de Doutorado, Worcester Polytechnic Institute, 265 páginas, 2005.
- BURGE, J. E.; BROWN, D. C. Rationale-Based Support for Software Maintenance. In: *A. H. Dutoit and R. McCall and I. Mistrík and B. Paech (eds.), Rationale Management in Software Engineering*, Springer, p. 273–296, 2006.
- BUSCHER, M.; CHRISTENSEN, M.; GRNBK, K.; KROGH, P.; MOGENSEN, P.; SHAPIRO, D.; ORBK, P. Collaborative Augmented Reality Environments: Integrating VR, Working Materials, and Distributed Work Spaces. In: *CVE '00: Proceedings of the Third International Conference on Collaborative Virtual Environments*, San Francisco, Estados Unidos: ACM Press, 2000, p. 47–56.
- CAGNIN, M. I.; PAIVA, D. M. B.; MALDONADO, J. C.; PENTEADO, R. D.; FORTES, R. P. M.; GERMANO, F. S. R. From Design Rationale to Reengineering Rationale: Lessons Learned in a Maintenance Pilot Case Study. In: *IV Jornadas Iberoamericanas*

- de Ingeniería del Software e Ingeniería del Conocimiento*, Madrid, Espanha, 2004, p. 61–66.
- CANFORA, G.; CASAZZA, G.; DE LUCIA, A. A Design Rationale Based Environment for Cooperative Maintenance. *International Journal of Software Engineering and Knowledge Engineering*, v. 10, n. 5, p. 627–645, 2000.
- CARLSON, C. G. The Dating Game, Survivor, and Other War Games: Interesting, Enjoyable and Effective Methods for Conducting Collaborative Research. In: *Proceedings of the 35th Hawaii International Conference on System Sciences*, Big Island, Estados Unidos: IEEE Computer Society, 2002.
- CAVALCANTI, M. C.; MATTOSO, M.; CAMPOS, M. L.; SIMON, E.; LLIRBAT, F. An Architecture for Managing Distributed Scientific Resources. In: *14th International Conference on Scientific and Statistical Database Management*, Edinburgh, Reino Unido: IEEE Computer Society, 2002.
- CAVIN, R. K.; SUMNEY, L. W.; BURGER, R. M. The Semiconductor Research Corporation: Cooperative Research. *Proceedings of the IEEE*, v. 77, n. 9, p. 1327–1344, 1989.
- CERVO, A.; BERVIAN, P. A. *Método Científico*. 3 ed. MacGraw-Hill, 1983.
- CHANG, C. K.; TRUBOW, G. B. Joint Software Research Between Industry and Academia. *IEEE Software*, v. 7, n. 6, p. 71–77, 1990.
- CHIAVENATO, I. *Recursos humanos*. 7 ed. Atlas, 2002.
- CHIROUZE, O.; CLEARY, D.; MITCHELL, G. G. A Software Methodology for Applied Research: eXtreme Researching. *Software – Practice and Experience*, v. 35, n. 15, p. 1441–1454, 2005.
- CHUNG, P.; GOODWIN, R. An Integrated Approach to Representing and Accessing Design Rationale. *Engineering Applications of Artificial Intelligence*, v. 11, n. 1, p. 149–159, 1998.
- CMMI Capability Maturity Model Integration, Version 1.2. 2006.
- COCCIA, M. Satisfaction, Work Involvement and R&D Performance. *International Journal Human Resources Development and Management*, v. 1, n. 2, p. 3–4, 2001.
- COCCIA, M. Metrics of R&D Performance and Management of Public Research Labs. In: *Engineering Management Conference*, 2003, p. 231–235.

- COLLIER, B.; DEMARCO, T.; FEAREY, P. A Defined Process for Project Postmortem Review. *IEEE Software*, v. 13, n. 4, p. 65–72, 1996.
- CONKLIN, J.; BEGEMAN, M. L. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, v. 6, n. 4, p. 303–331, 1988.
- CONKLIN, J.; BURGESS-YAKEMOVIC, K. A Process-Oriented Approach to Design Rationale. In: *Design Rationale Concepts, Techniques and Use*, T. Moran and J. Carrol (editors), Lawrence Erlbaum Associates, p. 293–428, 1996.
- CONN, R. A Reusable, Academic-strength, Metrics-based Software Engineering Process for Capstone Courses and Projects. In: *SIGCSE'04: Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, Estados Unidos: ACM Press, 2004, p. 492–496.
- COOK, C.; CHURCHER, N.; IRWIN, W. Towards Synchronous Collaborative Software Engineering. In: *Proceedings of the 11th Asia-Pacific Software Engineering Conference*, Busan, Korea: IEEE Computer Society, 2004, p. 230–239.
- CUBRANIC, D.; MURPHY, G. C. Hipikat: Recommending Pertinent Software Development Artifacts. In: *Proceedings of the 25th International Conference on Software Engineering*, Portland, Oregon: IEEE Computer Society, 2003, p. 408–418.
- CUGOLA, G.; NITTO, E. D.; FUGGETTA, A. The JEDI Event-Based Infrastructure and its Application to the Development of the OPSS WFMS. *IEEE Transactions on Software Engineering*, v. 27, n. 9, p. 827–850, 2001.
- DANNELLY, R. S.; STEIDLEY, C. W. A National Science Foundation Supported Undergraduate CSMET Research Project. *Journal of Computing Sciences in Colleges*, v. 17, n. 5, p. 13–17, 2002.
- DAVENPORT, T. H. *Reengenharia de Processos*. 2 ed. Campus, 1994.
- DESOUZA, K. C.; DINGSOYR, T.; AWAZU, Y. Experiences with Conducting Project Postmortems: Reports versus Stories. *Software Process: Improvement and Practice*, v. 10, n. 2, p. 203–215, 2005.
- DURSCKI, R. C.; SOEKE, M. E.; BUNET, R. C.; REINEHR, S. S. Processo de Desenvolvimento para Projetos Tecnológicos: Uma Abordagem Baseada em Survey. In: *XXIV Encontro Nacional de Engenharia de Produção*, Florianópolis - SC, 2004, p. 5443–5450.

- DUSTDAR, S. CARAMBA - A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams. *Distributed and Parallel Databases*, v. 15, n. 1, p. 45–66, 2004.
- DUTOIT, A. H.; MCCALL, R.; MISTRÍK, I.; PAECH, B. Rationale Management in Software Engineering: Concepts and Techniques. In: *A. H. Dutoit and R. McCall and I. Mistrík and B. Paech (eds.), Rationale Management in Software Engineering*, Springer, p. 1–48, 2006.
- DUTOIT, A. H.; PAECH, B. Rationale Management in Software Engineering. In: *Handbook of Software Engineering and Knowledge Engineering*, World Scientific Publishing Company, p. 787–816, 2000.
- EBERT, C.; NEVE, P. D. Surviving Global Software Development. *IEEE Software*, v. 18, n. 2, p. 62–69, 2001.
- ELLIS, H. J. C.; MEAD, N. R.; MORENO, A. M.; SEIDMAN, S. B. Industry/University Software Engineering Collaborations for the Successful Reeducation of Non-Software Professionals. In: *Proceedings of the 16th Conference on Software Engineering Education and Training*, Madrid, Espanha: ACM Press, 2003, p. 33–44.
- ESTEVES, R. P.; ABALÉM, A. J. G.; STANTON, M. A. Algorithm for Establishing LSPs for Multicast Communication over GMPLS-based Networks. In: *4th International Information and Telecommunication Technologies Symposium*, Florianópolis - SC, 2005.
- FALBO, R. A. *Integração de Conhecimento em um Ambiente de Desenvolvimento de Software*. Tese de Doutorado, Universidade Federal do Rio de Janeiro, 203 páginas, 1998.
- FALESSI, D.; CANTONE, G.; BECKER, M. Documenting Design Decision Rationale to Improve Individual and Team Design Decision Making: an Experimental Evaluation. In: *ISESE '06: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering*, Rio de Janeiro - RJ: ACM Press, 2006, p. 134–143.
- FAROOQ, U.; CARROLL, J. M.; GANOE, C. H. Supporting Creativity in Distributed Scientific Communities. In: *GROUP '05: Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work*, Sanibel Island, Estados Unidos: ACM Press, 2005, p. 217–226.
Disponível em <http://doi.acm.org/10.1145/1099203.1099242>
- FELLER, J.; FITZGERALD, B. A Framework Analysis of the Open Source Software Paradigm. In: *21st International Conference on Information Systems*, Brisbane, Australia, 2000, p. 58–69.

- FELTRIM, V. D. *Uma Abordagem baseada em Córpus e em Sistemas de Crítica para a Construção de Ambientes Web de Auxílio à Escrita Acadêmica em Português*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 181 páginas, 2004.
- FILHO, W. P. P. *Engenharia de Software – Fundamentos, Métodos e Padrões*. LTC, 2001a.
- FILHO, W. P. P. Requirements for an Educational Software Development Process. In: *Proceedings of The 6th Annual Conference on Innovation and Technology in Computer Science Education*, Canterbury, Inglaterra, 2001b.
- FIORINI, S. T. *Arquitetura para Reutilização de Processos de Software*. Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, 243 páginas, 2001.
- FISCHER, G.; MCCALL, R.; MORCH, A. Design Environments for Constructive and Argumentative Design. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'89)*, Austin, Estados Unidos, 1989, p. 269–275.
- FORTES, R. P. M.; FREIRE, A. P.; VIEIRA, V. H.; PAIVA, D. M. B. An Academic Web-Based Agenda and its Engineering Process. In: *VII Workshop Ibero Americano de Ingeniería de Requisitos Y Desarrollo de Ambientes de Software*, Arequipa, Peru, 2004a, p. 151–156.
- FORTES, R. P. M.; SILVA, M. A. G.; FREIRE, A. P.; JUNQUEIRA, D. C. SAFE – Software Engineering Available For Everyone. In: *V Workshop sobre Software Livre*, Porto Alegre - RS, 2004b, p. 203–206.
- FRANCISCO, S. D. *DocRationale: Uma Ferramenta para Suporte a Design Rationale de Artefatos de Software*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos - SP, 123 páginas, 2004.
- FRANKFORT-NACHMIAS, C.; NACHMIAS, D. *Research Methods in the Social Sciences*. 4 ed. St. Martin's Press, 1992.
- FREIRE, A. P. *Evolução da Ferramenta DocRat Visando Suporte a Design Rationale Ágil e Versões de Software. Monografia de Projeto de Conclusão de Curso*. Relatório Técnico, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 46 páginas, 2005.
- FREIRE, A. P.; PAIVA, D. M. B.; FORTES, R. P. M. Gerenciamento de Recursos Acadêmicos – um Estudo de Caso. *Revista Brasileira de Informática na Educação*, v. 12, n. 2, p. 78–85, 2004.

- FUGGETTA, A. Software Process: A Roadmap. In: *Proceedings of the Conference on the Future of Software Engineering*, Limerick, Irlanda: ACM Press, 2000, p. 25–34.
- GARCIA, V. C.; LUCRÉDIO, D.; PINTO, L. F. P.; ALVARO, A.; ALMEIDA, E. S.; PRADO, A. F. Uma Ferramenta CASE para o Desenvolvimento de Software Orientado a Aspectos. In: *Sessão de ferramentas do XVIII Simpósio Brasileiro de Engenharia de Software (SBES 2004)*, Brasília - DF, 2004.
- GEORGE, G. P.; PICARD, G.; GLEIZES, M. P.; GLIZE, P. Living Design for Open Computational Systems. In: *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Manchester, Inglaterra, 2003, p. 389–394.
- GIL, A. C. *Métodos e Técnicas da Pesquisa Social*. 5 ed. Atlas, 1999.
- GIL, A. C. *Como Elaborar Projetos de Pesquisa*. 4 ed. Atlas, 2002.
- GOES, J. A. *Perf: Ambiente de desenvolvimento e estimação temporal de sistemas de tempo real*. Dissertação de Mestrado, CEFET-PR, 123 páginas, 2001.
- GOES, J. A.; RENAUX, D. P. B. PERF: Um Ambiente de Desenvolvimento Voltado para Testes Temporais de Software. In: *XV Simpósio Brasileiro de Engenharia de Software, XVIII Sessão de Ferramentas*, 2001, p. 404–409.
- GOLDENBERG, M. *A Arte de Pesquisar*. 7 ed. Record, 2003.
- GONÇALVES, H. A. *Manual de Monografia, Dissertação e Tese*. 1 ed. AVERCAMP, 2004.
- GONZÁLEZ-BARAHONA, J. M.; ROBLES, G. Free Software Engineering: A Field to Explore. *UPGRADE*, v. 4, n. 4, p. 49–54, 2003.
- GREMBA, J.; MYERS, C. The IDEAL Model: A Practical Guide for Improvement. *Software Engineering Institute (SEI) publication*, , n. 3, <http://www.sei.cmu.edu/ideal/ideal.bridge.html>, 1997.
- GRUBER, T. R.; RUSSEL, D. M. *Design Knowledge and Design Rationale: A Framework for Representation, Capture, and Use*. Technical Report KSL 90-45, Knowledge Systems Laboratory, Standford, California, 40p., 1991.
- GUAN, L. Z.; BERZINS, V.; ZHANG, L.; FLOODEEN, D.; V, C.; PUETT, J.; BROWN, M. Requirements-document-based Prototyping of CARA Software. *International Journal on Software Tools for Technology Transfer*, v. 5, n. 4, p. 370–390, 2004.

- GUPTA, A.; S.VADHAVKAR; PEÑA-MORA, F.; YEUNG, J. An Integrated Framework to Support Distributed CAD over the Internet. In: *International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2001.
- HAGGE, L.; HOUDEK, F.; LAPPE, K.; PAECH, B. Using Patterns for Sharing Requirements Engineering Process Rationales. In: *A. H. Dutoit and R. McCall and I. Mistrík and B. Paech (eds.), Rationale Management in Software Engineering*, Springer, p. 409–427, 2006.
- HAGSAND, O. Interactive Multiuser Virtual Environments in the DIVE System. *IEEE Multimedia*, v. 3, n. 1, p. 30–39, 1996.
- HARRIS, G.; KAINE, G. The Determinants of Research Performance: a Study of Australian University Economists. *Higher Education*, , n. 27, p. 191–201, 1994.
- HAWRYSH, S. P.; RUPRECHT, J. Light Methodologies: Its Like Déjà Vu All Over Again. *Cutter IT Journal*, v. 13, n. 11, p. 4–12, 2000.
- HAWTHORNE, M. J.; PERRY, D. E. Software Engineering Education in the Era of Outsourcing, Distributed Development and Open Source Software: Challenges and Opportunities. In: *27th International Conference on Software Engineering*, St. Louis, Estados Unidos, 2005, p. 643–644.
- HAYNES, S. R. Three Studies of Design Rationale as Explanation. In: *A. H. Dutoit and R. McCall and I. Mistrík and B. Paech (eds.), Rationale Management in Software Engineering*, Springer, p. 53–71, 2006.
- HERBSLEB, J. D.; PAULISH, D. J.; M. Global Software Development at Siemens: Experience from Nine Projects. In: *27th International Conference on Software Engineering*, St. Louis, Estados Unidos, 2005, p. 524–533.
- HIGHSMITH, J.; COCKBURN, A. Agile Software Development: the Business of Innovation. *IEEE Computer*, v. 34, n. 9, p. 120–127, 2001.
- HILBURN, T. B. Teams Need a Process! In: *ITiCSE '00: Proceedings of the 5th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finlândia: ACM Press, 2000, p. 53–56.
- HOHN, E. N. *Técnicas de Leitura de Especificação de Requisitos de Software: Estudos Empíricos e Gerência de Conhecimento em Ambientes Acadêmico e Industrial*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 219 páginas, 2003.

- HOLMES, R.; MURPHY, G. C. Using Structural Context to Recommend Source Code Examples. In: *27th International Conference on Software Engineering*, St. Louis, Estados Unidos, 2005.
- HULJENIC, D.; DESIC, S.; MATIJASEVIC, M. Project Management in Research Projects. In: *8th International Conference on Telecommunications – ConTEL 2005*, Nis, Serbia, 2005, p. 663–669.
- HUMPHREY, W. S. *A Discipline for Software Engineering*. Addison-Wesley Publishing Company, 1995.
- HUMPHREY, W. S. *Introduction to the Team Software Process*. Addison-Wesley Publishing Company, 1999.
- HUMPHREY, W. S.; KELLNER, M. I. Software Process Modeling: Principles of Entity Process Models. In: *11th International Conference on Software Engineering*, 1989, p. 331–342.
- HWANG, Y. H.; PARK, J. G. Approaches and Requirements to Develop and Improve the Standard Processes for a Research and Development Organization. *Systems Engineering*, v. 9, n. 1, p. 35–43, 2006.
- IEEE STANDARD 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. 2000.
- IGEL, B.; NUMPRASERTCHAI, S. Knowledge Management in University R&D in Thailand. In: *Proceedings of IEEE International Engineering Management Conference*, Singapura: IEEE Press, 2004, p. 463–467.
- IONESCU, B.; IONESCU, M.; VERES, S.; IONESCU, D. Design and Architecture of an IP over Optical and Optical Ethernet Next Generation Network – NCIT* net2. In: *4th International Information and Telecommunication Technologies Symposium*, Florianópolis - SC, 2005.
- ISO 9001 ISO 9001 Quality Systems – Model for Quality Assurance in Design, Development, Production, Installation and Servicing. 1994.
- ISO/IEC 12207 ISO/IEC 12207 Information technology – software life-cycle processes. 1995.
- ISO/IEC 14598 ISO/IEC 14598: Information Technology, Software Product Evaluation. 1996.
- ISO/IEC 15288 ISO/IEC 15288 Systems Engineering – System Life Cycle Processes. 2002.

- ISO/IEC 9126 ISO/IEC 9126 Software Engineering – Product Quality, Parts 1–4. 2002.
- JACCHERI, M. L.; CONRADI, R. Techniques for process model evolution in epos. *IEEE Transactions on Software Engineering*, v. 19, n. 12, p. 1145–1156, 1993.
- JENKINS, M. PRO-SOFTWARE: A Government-Industry-Academia Partnership that Worked . In: *17th Conference on Software Engineering Education and Training (CSEET'04)*, Norfolk, Estados Unidos: IEEE Computer Society, 2004, p. 92–97.
- JUBILEU, A. P. *Aquisição de Conhecimento Como Apoio ao Método de Engenharia Reversa Fusion-RE/I*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 141 páginas, 1999.
- JUNIOR, M. G. C. *Um Serviço de Coordenação de Processos Integrados ao Ambiente de Engenharia de Software E-WebProject*. Dissertação de Mestrado, Instituto Nacional de Pesquisas Espaciais, 274 páginas, 2004.
- JUNIOR, M. G. C.; SANT'ANNA, N.; FILHO, L. F. B.; GENVIGIR, E. C.; LUQUE, L.; TAVARES, R. P.; CASILLO, B. H. (PDE) Process Definition Environment: Uma Ferramenta para Apoio à Definição de Processos de Software no Ambiente e-WebProject. In: *XIV Congresso Internacional de Tecnologia de Software*, Curitiba - PR, 2003, p. 42–55.
- KAMMER, P. J.; BOLCER, G. A.; TAYLOR, R. N.; HITOMI, A. S. Supporting Distributed Workflow Using HTTP. In: *Proceedings of the International Conference on Software Engineering*, Kyoto, Japão, 1998.
- KARSENTY, L. An Empirical Evaluation of Design Rationale Documents. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96)*, Vancouver, Canadá, 1996, p. 13–18.
- KATO, Y.; HORI, K.; TAKETA, K. Capturing Design Rationale by Annotating E-mails. In: *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Estados Unidos, 2002.
- KELLNER, M. I. Representation Formalisms for Software Process Modelling. In: *Proceedings of the 4th International Software Process Workshop on Representing and Enacting the Software Process*, Devon, Reino Unido: ACM Press, 1988, p. 93–96.
- KIM, W.; SUH, Y.; WHINSTON, A. B. An IBIS and Object-Oriented Approach to Scientific Research Data Management. *Journal of Systems and Software*, v. 23, n. 2, p. 183–197, 1993.

- KIRCHER, M.; JAIN, P.; CORSARO, A.; LEVINE, D. Distributed eXtreme Programming. In: *Proceedings of the Second International Conference on eXtreme Programming (XP 2001)*, 2001, p. 66–71.
- KITCHENHAM, B. *Procedures for Performing Systematic Reviews*. Technical Report TR/SE-0401, 28 páginas, Software Engineerign Group – Department of Computer Science, ISSN 1353-7776, 2004.
- KITCHENHAM, B.; PFLEEGER, S. L. Software Quality: the Elusive Target. *IEEE Software*, v. 13, n. 1, p. 12–21, 1996.
- KOCHUT, K.; ARNOLD, J.; SHETH, A.; MILLER, J.; KRAEMER, E.; ARPINAR, B.; CARDOSO, J. IntelliGEN: A Distributed Workflow System for Discovering Protein-Protein Interactions. *Distributed and Parallel Databases*, v. 13, n. 1, p. 43–72, 2003.
- KOMI-SIRVIO, S.; TIHINEN, M. Lessons Learned by Participants of Distributed Software Development. *Knowledge and Process Management*, v. 12, n. 2, p. 108–122, 2005.
- KORNECKI, A.; KHAJENOORI, S.; GLUCH, D.; KAMELI, N. On a Partnership Between Software Industry and Academia. In: *Proceedings of the 16th Conference on Software Engineering Education and Training*, Madrid, Espanha, 2003, p. 60–69.
- KOSCIANSKI, A.; SOARES, M. S. *Qualidade de Software*. 1 ed. Novatec, 2006.
- KOTONYA, G.; SOMMERVILLE, I. *Requirements Engineering: Processes and Techniques*. John Wiley, 1998.
- KOTTING, B.; MAURER, F. Approaching Software Support for Virtual Software Corporations. In: *21th International Conference on Software Engineering*, Los Angeles, Estados Unidos, 1999.
- KOUZES, R. T.; MYERS, J. D.; W. A WULF Collaboratories: Doing Science on the Internet. *IEEE Computer*, v. 29, n. 8, p. 40–46, 1996.
- KRISHMAN, M. S.; MUKHOPADHYAY, T.; ZUBROW, D. Software Process Models and Project Performance. *Information Systems Frontiers*, v. 1, n. 3, p. 267–277, 1999.
- KUNZ, W.; RITTEL, W. Issues as Elements of Information Systems. *Working paper 131*, Center for Planning and Development Research, University of California, Berkeley, 1970.
- LANZA, M.; DUCASSE, S.; GALL, H.; PINZGER, M. CodeCrawler: an Information Visualization Tool for Program Comprehension. In: *Proceedings of the 27th International Conference on Software Engineering*, St. Louis, Estados Unidos: ACM Press, 2005, p. 672–673.

- LARA, S. M. A.; FREIRE, A. P.; FORTES, R. P. M. *DocRat: Produto da Re-engenharia da Ferramenta DocRationale*. Relatório Técnico, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 72 páginas, 2005.
- LAVAZZA, L.; BARRESI, G. Automated Support for Process-aware Definition and Execution of Measurement Plans. In: *Proceedings of the 27th International Conference on Software Engineering*, St. Louis, Estados Unidos, 2005, p. 234–243.
- LAVILLE, C.; DIONNE, J. *A Construção do Saber: Manual de Metodologia da Pesquisa em Ciências Humanas*. 1 ed. UFMG, 1999.
- LEE, J. Design Rationale Systems: Understanding the Issues. *IEEE expert/Intelligent Systems and Their Applications*, v. 12, n. 3, p. 78–85, 1997.
- LOUGHER, R.; RODDEN, T. Group Support for the Recording and Sharing of Maintenance Rationale. *Software Engineering Journal*, v. 8, n. 6, p. 295–306, 1993.
- LU, S. C.-Y.; UDWADIA, F.; BURKETT, B.; CAI, J. Design Rationale for Collaborative Engineering Design in the Socio-Technical Framework. 1999.
Disponível em <http://impact.usc.edu/cer1/Task2/task2-2-3/DesignRationale.pdf>
(Último Acesso em 03/03/2002)
- LUCRÉDIO, D. *Extensão da Ferramenta MVCASE com Serviços Remotos de Armazenamento e Busca de Artefatos de Software*. Dissertação de Mestrado, Universidade Federal de São Carlos, 104 páginas, 2005.
- LUCRÉDIO, D.; ALMEIDA, E. S.; BIANCHINI, C. P.; PRADO, A. F.; TREVELIN, L. C. Orion - A Component-Based Software Engineering Environment. *Journal of Object Technology*, v. 3, n. 4, p. 51–74, 2004.
- LUCRÉDIO, D.; ALMEIDA, E. S.; PRADO, A. F. MVCASE – Eight years of Experiences with a CASE Tool. Submetido ao International Journal on Software Tools Technology Transfer, em revisão, 2006.
- LUWEL, M.; NOYONS, C. M.; MOED, F. Bibliometric Assessment of Research Performance in Flanders: Policy Background and Implications. *R&D Management*, v. 29, n. 2, p. 133–141, 1999.
- MACKE, S.; NEW, J.; COXON, J.; MANENTE, B.; KHAJENOORI, S.; HIRMANPOUR, I.; CEBERIO, A. An Industry/Academic Partnership that Worked: an in Progress Report. In: *Ninth Conference on Software Engineering Education*, Daytona Beach, Estados Unidos: IEEE Computer Society, 1996, p. 234–245.

- MACLEAN, A.; YOUNG, R. M.; BELLOTTI, V.; MORAN, T. P. Questions, Options and Criteria: Elements of Design Space Analysis. *Human-Computer Interaction*, v. 6, n. 3–4, p. 201–250, 1991.
- MADHAVJI, N. H. The Process Cycle. *Software Engineering Journal*, v. 6, n. 5, p. 234–242, 1991.
- MAIDANTCHIK, C. L. L. *Gerência de Processos de Software para Equipes Geograficamente Dispersas*. Tese de Doutorado, COPPE, Universidade Federal do Rio de Janeiro, 214 páginas, 1999.
- MALDONADO, J. C.; CARVER, J.; SHULL, F.; FABBRI, S.; DÓRIA, E.; MARTIMIANO, L.; MENDONÇA, M.; BASILI, V. Perspective-Based Reading: A Replicated Experiment Focused on Individual Reviewer Effectiveness. *Empirical Software Engineering*, v. 11, n. 1, p. 119–142, 2006.
Disponível em <http://www.springerlink.com/content/9543346630211402/>
- MARCONI, M. A.; LAKATOS, E. M. *Metodologia do Trabalho Científico*. 6 ed. Atlas, 2001.
- MARCONI, M. A.; LAKATOS, E. M. *Técnicas de Pesquisa*. 5 ed. Atlas, 2002.
- MARTIN, J. C. Formal Methods Software Engineering for the CARA System. *International Journal on Software Tools for Technology Transfer*, v. 5, n. 4, p. 301–307, 2004.
- MAURER, F.; KAISER, G. Software Engineering in the Internet Age. *IEEE Internet Computing*, v. 2, n. 5, p. 22–24, 1998.
- MCCALL, R. J. PHI: A Conceptual Foundation for Design Hypermedia. *Design Studies*, v. 12, n. 1, p. 30–41, 1991.
- MCCAULEY, R. Agile Development Methods Poised to Upset Status Quo. *SIGCSE Bulletin*, v. 33, n. 4, p. 14–15, 2001.
- MIYOSHI, T. Early Experience with Software Process Assessment using SPICE Framework at Software Research Associates. *Software Process: Improvement and Practice*, v. 2, n. 3, p. 211–235, 1998.
- MONK, S.; SOMMERVILLE, I.; PENDARIES, J. M.; DURIN, B. Supporting Design Rationale For System Evolution. In: *Proceedings of the Fifth European Software Engineering Conference*, Sitges, Espanha, 1995.

- MORAN, T. P.; CARROLL, J. M. *Design Rationale: Concepts, Techniques, and Use Computers, Cognition, and Work*. New Jersey: Lawrence Erlbaum Associates, 659 páginas, 1996.
- MOREIRA, M. R. S.; SILVA, R. F. V.; ELLERY, C. E. M.; AQUINO, J. F. S.; COSTA, T. C.; RODRIGUES, M. A. A.; ANDRADE, R. M. C. WQoS: Um Framework de Alto Nível para Provisão de QoS em Sistemas Móveis sem Fio. In: *4th International Information and Telecommunication Technologies Symposium*, Florianópolis - SC, 2005, p. 117–127.
- MOURA, M. V. *No-Risk Planning – um Sistema de Suporte ao Trabalho em Grupo: uma Proposta para Web*. Relatório Técnico, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 53 páginas, 2001.
- MYERS, K. L.; ZUMEL, N. B.; GARCIA, P. Automated Capture of Rationale for the Detailed Design Process. In: *Proceedings of the Eleventh Conference on Innovative Applications of Artificial Intelligence*, Orlando, Estados Unidos, 1999, p. 876–883.
- NGUYEN, L.; SWATMAN, P. A. Promoting and Supporting Requirements Engineering Creativity. In: *A. H. Dutoit and R. McCall and I. Mistrík and B. Paech (eds.), Rationale Management in Software Engineering*, Springer, p. 209–230, 2006.
- NAZI, M.; WILSON, D.; ZOWGHI, D. Critical Success Factors for Software Process Improvement Implementation: An Empirical Study. *Software Process: Improvement and Practice*, v. 11, n. 2, p. 193–211, 2006.
- NUMPRASERTCHAI, S.; IGEL, B. Managing Knowledge Through Collaboration: Two Cases of Managing Research in University Laboratories in Thailand. In: *Engineering Management Conference, IEMC 2003*, Albany, Estados Unidos: IEEE Press, 2003, p. 575–579.
- NUNAMAKER, J. F.; CHEN, M. Systems Development in Information Systems Research. In: *Proceedings of the Twenty-Third Annual Hawaii International Conference*, IEEE, 1990, p. 631–640.
- OLIVEIRA, H. L. R.; ROCHA, C. R. P.; GONÇALVES, K. M.; SOUZA, C. R. B. Utilização de Sistemas Críticos nas Atividades de Engenharia de Domínio e de Aplicações. In: *XV Simpósio Brasileiro de Engenharia de Software*, Rio de Janeiro - RJ, 2001, p. 21–35.
- OLIVEIRA, K. R. *Adequas: Ferramenta Fuzzy para Avaliação da Qualidade de Software*. Dissertação de Mestrado, Universidade de Fortaleza, 130 páginas, 2002.

- OLIVEIRA, K. R.; BELCHIOR, A. D. Adequas: Ferramenta Fuzzy para Avaliação da Qualidade de Software. In: *XVII Simpósio Brasileiro de Engenharia de Software, X Sessão de Ferramentas*, Manaus - AM, 2003, p. 31–36.
- OLSON, G. M.; TEASLEY, S.; BIETZ, M. J.; COGBURN, D. L. Collaboratories to Support Distributed Science: the Example of International HIV/AIDS Research. In: *SAICSIT '02: Proceedings of the 2002 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement Through Technology*, Port Elizabeth, South Africa, 2002, p. 44–51.
- OPEN SOURCE INITIATIVE The Open Source Definition. Visitado em Julho de 2006, 2006.
Disponível em <http://www.opensource.org/docs/definition.php>
- O'REILLY, T. Lessons from Open-Source Software Development. *Communications of ACM*, v. 42, n. 4, p. 32–37, 1999.
- PAASVAARA, M.; LASSENIUS, C. Collaboration Practices in Global Inter-organizational Software Development Projects. *Software Process: Improvement and Practice*, v. 8, n. 4, p. 183–199, 2004.
- PAIVA, D. M. B.; FORTES, R. P. M. Design Rationale in Software Engineering: A Case Study. In: *Seventeenth International Conference on Software Engineering and Knowledge Engineering*, Taipei, Taiwan, 2005, p. 342–348.
- PAIVA, D. M. B.; FREIRE, A. P.; FORTES, R. P. M. Design Rationale in Academic Software Development: Requirements for a Representation Model. In: *Eighteenth International Conference on Software Engineering and Knowledge Engineering – SEKE 2006*, San Francisco, 2006a, p. 469–472.
- PAIVA, D. M. B.; FREIRE, A. P.; LUCRÉDIO, D.; BRAGA, R. T. V.; FORTES, R. P. M. Reinforcing Design Rationale in Software Projects Developed in Academic Environment. *International Transactions on Systems Science and Applications*; Aceito para publicação, 2007.
- PAIVA, D. M. B.; FREIRE, A. P.; SANCHES, R.; FORTES, R. P. M. Definindo, Implantando and Melhorando Processos de Software em Ambiente Acadêmico. In: *VI Simpósio Internacional em Melhoria de Processo de Software – SIMPROS*, São Paulo - SP, 2004, p. 71–82.
- PAIVA, D. M. B.; LUCRÉDIO, D.; FORTES, R. P. M. MVCASE – Incluindo Design Rationale para Auxílio à Modelagem em Projetos de Pesquisa. In: *XIII Sessão de Ferramentas – XX Simpósio Brasileiro de Engenharia de Software*, Florianópolis - SC, 2006b, p. 73–78.

- PAIVA, D. M. B.; MANDUCA, A. M.; SILVA, M. A. G.; QUEMELLO, L. J.; BRAGA, R. T. V.; FORTES, R. P. M. Reforçando a Comunicação com Uso de uma Ferramenta de Software Livre em Ensino de Engenharia de Software. In: *XIII Workshop sobre Educação em Computação, XXV Congresso da Sociedade Brasileira de Computação*, São Leopoldo - RS, 2005.
- PAIVA, D. M. B.; SANCHES, R.; FORTES, R. P. M. Melhoria de Processo de Software no Contexto do Desenvolvimento Acadêmico de uma Agenda na Web. In: *XIV Congresso Internacional de Tecnologia de Software (CITS 2003)*, Curitiba - PR, 2003, p. 56–68.
- PAIVA, D. M. B.; TURINE, M. A. S.; FORTES, R. P. M. Processo de Software Livre em Ambiente Acadêmico: Experiências e Lições Aprendidas. In: *XXXII Latin-American Conference on Informatics – CLEI 2006*, Santiago, Chile, 2006c.
- PALMER, S. R.; FELSING, J. M. *A Practical Guide to Feature-Driven Development*. Upper Saddle River, Prentice Hall, 2002.
- PANOSSO, G. C. *Informações de Design Rationale como Auxílio à Manutenção de Software*. Relatório técnico, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos-SP, 36 páginas, 2004.
- PANSANATO, L. T. E. *Autoria de Aplicações Hiperídia para Ensino*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 1999.
- PARNAS, D.; CLEMENTS, P. A Rationale Design Process: How and Why to Fake it. *IEEE Transactions of Software Engineering*, v. 12, n. 2, p. 251–257, 1986.
- PAULK, M. C.; WEBER, C. V.; CURTIS, B.; CHRISISS, M. B. *The Capability Maturity Model: Guidelines for Improving the Software Process/CMU/SEI*. Addison-Wesley Publishing Company, 1995.
- PÁDUA, E. M. M. *Metodologia da Pesquisa: Abordagem Teórico-Prática*. 6 ed. Papirus, 2000.
- PERPICH, J. M.; PERRY, D. E.; PORTER, A. A.; VOTTA, L. G.; WADE, M. W. Anywhere, Anytime Code Inspections: Using the Web to Remove Inspection Bottlenecks in Large-scale Software Development. In: *ICSE97: Proceedings of the 19th International Conference on Software Engineering*, Boston, Estados Unidos: ACM Press, 1997, p. 14–21.
- PFLEEGER, S. L. *Software Engineering: Theory and Practice*. 2 ed. New Jersey: Prentice Hall, 659 p., 2001.

- PIMENTEL, M. G. C. MAPHE: Metodologia de Apoio a Projetos de Hiperdocumentos Educacionais. In: *VIII Simpósio Brasileiro de Informática na Educação (SBIE97)*, São José dos Campos - SP, 1997, p. 351–368.
- PINHEIRO, M. K.; LIMA, J. V.; BORGES, M. R. S. A Framework for Awareness Support in Groupware Systems. In: *7th International Conference on Computer Supported Cooperative Work in Design*, Rio de Janeiro - RJ, 2002, p. 13–18.
- POTTS, C.; BRUNS, G. Recording the Reasons for Design Decisions. In: *Proceedings of the International Conference on Software Engineering*, Kyoto, Japão: IEEE CS Press, 1988, p. 418–427.
- PRADO, A. F.; LUCRÉDIO, D. MVCASE: Ferramenta CASE Orientada a Objetos. In: *Sessão de ferramentas do XIV Simpósio Brasileiro de Engenharia de Software (SBES 2000)*, João Pessoa - PB, 2000.
- PRADO, A. F.; LUCRÉDIO, D. Ferramenta MVCASE - Estágio Atual: Especificação, Projeto e Construção de Componentes. In: *Sessão de ferramentas do XV Simpósio Brasileiro de Engenharia de Software (SBES 2001)*, Rio de Janeiro - RJ, 2001, p. 368–373.
- PRESSMAN, R. S. *Engenharia de Software*. 5 ed. McGraw Hill, 2001.
- PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. 6 ed. McGraw Hill, 2005.
- PRIKLADNICKI, R.; AUDY, J. L. N.; EVARISTO, R. Global Software Development in Practice Lessons Learned. *Software Process: Improvement and Practice*, v. 8, n. 4, p. 267–281, 2004.
- PROJECT MANAGEMENT INSTITUTE Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos – Guia PMBOK. 2004.
- QUEMELLO, L. J.; MANDUCA, A. M.; FORTES, R. P. M. *Uma Introdução ao Bugzilla*. Relatório Técnico 253, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2005.
- RAYMOND, E. S. The Cathedral and The Bazaar. In: *The Cathedral and The Bazaar*, 1st ed, Sebastopol: O'Reilly and Associates, p. 27–78, 1999.
- RAYMOND, E. S. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O' Reilly, 2001.

- REGLI, W. C.; HU, X.; ATWOOD, M.; SUN, W. A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval. *Engineering with Computers: An International Journal for Simulation-Based Engineering, special issue on Computer Aided Engineering in Honor of Professor Steven J. Fenves*, v. 16, p. 209–235, 2000.
- REIS, C. A. L. *Uma Abordagem Flexível para Execução de Processos de Software Evolutivos*. Tese de Doutorado, Universidade Federal do Rio Grande do Sul, 2003a.
- REIS, C. R. *Caracterização de um Processo de Software para Projetos de Software Livre*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, 247 páginas, 2003b.
- REIS, R. Q. *Uma Proposta de Suporte ao Desenvolvimento Cooperativo de Software no Ambiente PROSOFT*. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, 1998.
- RIBEIRO, T. M. *Evolução de uma Agenda para Grupos na Web*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, 81 páginas, 2003.
- RIBEIRO, T. M.; FORTES, R. P. M.; FREIRE, A. P. *Documentação do software agenda No Risk Planning*. Relatório Técnico 182, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2003.
- ROBILLARD, P. N.; ROBILLARD, M. Improving Academic Software Engineering Projects: A Comparative Study of Academic and Industry Projects. *Annals of Software Engineering*, v. 6, p. 343–363, 1998.
- ROCHA, A. R.; MONTONI, M.; SANTOS, G.; MAFRA, S.; FIGUEIREDO, S.; BESSA, A. Estação TABA: Uma Infra-estrutura para Implantação do Modelo de Referência para Melhoria de Processo de Software. In: *IV Simpósio Brasileiro de Qualidade de Software (SBQS'05)*, Porto Alegre - RS, 2005, p. 49–60.
- ROCHA, A. R. C.; MAIDANTCHIK, C.; OLIVEIRA, K.; TRAVASSOS, G. T.; XEXÉO, G. Experiência em Definição, Uso e Melhoria de Processos de Software. In: *A. R. C. Rocha and J. C. Maldonado and K. C. Weber (organizadores), Qualidade de Software, Teoria e Prática*, Prentice Hall, p. 185–191, 2001a.
- ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. *Qualidade de Software: Teoria e Prática*. 1 ed. São Paulo: Person Education do Brasil/Makron Books, 2001b.
- ROGERS, E. M.; CARAYANNIS, E. G.; KURIHARA, K.; ALLBRITTON, M. M. Cooperative Research and Development Agreements (CRADAs) as Technology Transfer Mechanisms. *R&D Management*, v. 28, n. 2, p. 79–88, 1998.

- ROMBACH, D. Experimentation: Engine for Applied Research and Technology Transfer in Software Engineering. Disponível em: http://sel.gsfc.nasa.gov/website/sew/1999/topics/rombach_SEW99paper.pdf; Acessado em 11/02/2007, 1999.
- ROMBACH, D. Fraunhofer: the german model for applied research and technology transfer. In: *Proceedings of the 22nd International Conference on Software Engineering*, Limerick, Irlanda: ACM Press, 2000, p. 531–537.
- ROSHANDEL, R.; HOEK, A. V. D.; MIKIC-RAKIC, M.; MEDVIDOVIC, N. MAE – A System Model and Environment for Managing Architectural Evolution. *ACM Transactions on Software Engineering and Methodology*, v. 13, n. 2, p. 240–276, 2004.
- ROST, J. Software Engineering Theory in Practice. *IEEE Software*, v. 22, n. 2, p. 94–96, 2005.
- SALOMON, D. V. *Como Fazer uma Monografia*. 11 ed. Martins Fontes, 2004.
- SANTE, D. G. *AutorE: Suportando Autoria Evolucionária em Ambientes de Captura*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 95 páginas, 2003.
- SANTOS, A. R. *Metodologia Científica: a Construção do Conhecimento*. 6 ed. DP&A, 2004.
- SCACCHI, W.; FELLER, J.; FITZGERALD, B.; HISSAM, S.; LAKHANI, K. Understanding Free/Open Source Software Development Processes. *Software Process: Improvement and Practice*, v. 11, n. 2, p. 95–105, 2006.
- SCHNEIDER, J.-G.; JOHNSTON, L. eXtreme Programming at Universities: an Educational Perspective. In: *Proceedings of the 25th International Conference on Software Engineering*, Portland, Oregon: IEEE Computer Society, 2003, p. 594–599.
- SCHNEIDEWIND, N. F. The State of Software Maintenance. *IEEE Transactions of Software Engineering*, v. 13, n. 3, p. 344–354, 1987.
- SCHWABER, K.; BEEDLE, M. *Agile Software Development with Scrum*. Upper Saddle River, Prentice Hall, 2002.
- SEGAL, J. When Software Engineers Met Research Scientists: A Case Study. *Empirical Software Engineering*, v. 10, n. 4, p. 517–536, 2005.
- SEVERINO, A. J. *Metodologia do Trabalho Científico*. 22 ed. Cortez, 2002.

- SHIPMAN, F.; MCCALL, R. Integrating Different Perspectives on Design Rationale: Supporting the Emergence of Design Rationale from Design Communication. *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, v. 11, n. 2, p. 141–154, 1997.
- SHULL, F.; CARVER, J.; TRAVASSOS, G. H. An Empirical Methodology for Introducing Software Processes. In: *ESEC/FSE-9: Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, Vienna, Austria: ACM Press, 2001, p. 288–296.
- SILVA, L. F.; SAYÃO, M.; LEITE, J. C. S. P.; BREITMAN, K. K. Enriquecendo o Código com Cenários. In: *XVII Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas*, Manaus - AM, 2003, p. 161–176.
- SMITH, A. J. The Task of the Referee. *IEEE Computer*, v. 23, n. 4, p. 65–71, 1990.
- SODRE, R. A. *Infra-estrutura de Segurança para Sistemas Distribuídos Baseada em Linguagem Formal*. Dissertação de Mestrado, Universidade de Brasília, 2001.
- SOFTWARE ENGINEERING STANDARDS COMMITTEE OF THE IEEE COMPUTER SOCIETY IEEE Recommended Practice for Software Requirements Specifications. 1998.
- SOMMERVILLE, I. *Software Engineering*. 6 ed. Addison Wesley, 2003.
- SONNENWALD, D. H.; WHITTON, M. C.; MAGLAUGHLIN, K. L. Evaluating a Scientific Collaboratory: Results of a Controlled Experiment. *ACM Transactions on Computer-Human Interaction*, v. 10, n. 2, p. 150–176, 2003.
- SOUZA, C.; SANTOS, D. B.; DIAS, K. L.; WAINER, J. A Model and Tool for Semi-Automatic Recording of Design Rationale in Software Diagrams. In: *Proceedings of the String Processing and Information Retrieval Symposium and International Workshop on Groupware*, Cancun, México: IEEE Computer Society, 1999, p. 306–313.
- SOUZA, O. R. *Processos de Apoio ao Desenvolvimento de Aplicações Web*. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, 2005.
- SPINDOLA, B.; ALBUQUERQUE, C.; MASCENA, J.; COELHO, K.; ALBUQUERQUE, R.; ROCHA, T. Definição e Melhoria de Processos em uma Fábrica de Software Livre. In: *VI Simpósio Internacional de Melhoria de Processos de Software*, São Paulo - SP, 2004, p. 189–200.

- STUTT, A.; MOTTA, E. Recording the Design Decisions of a Knowledge Engineering Community to Facilitate Re-use of Design Models. In: *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW'95)*, Banff, Canadá, 1995.
- SUCCI, G.; ZULIANI, P. Exploiting the Collaboration between Open Source Developers and Research. In: *4th Workshop on Open Source Software Engineering, 26th International Conference on Software Engineering*, Scotland, Reino Unido, 2004, p. 97–99.
- SUTCLIFFE, A. G.; RYAN, M. Experience with SCRAM, a Scenario Requirements Analysis Method. In: *Proceedings of the 3rd International Conference on Requirements Engineering*, Colorado, Estados Unidos: IEEE Computer Society, 1998, p. 164–171.
- TANG, A.; BABAR, M. A.; GORTON, I.; HAN, J. A Survey of the Use and Documentation of Architecture Design Rationale. In: *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture*, Pittsburgh, Estados Unidos: IEEE Computer Society, 2005, p. 89–98.
- TEUFEL, S.; CARLETTA, J.; MOENS, M. An Annotation Scheme for Discourse-level Argumentation in Research Articles. In: *Proceedings of the Ninth Meeting of the European Chapter of the Association for Computational Linguistics (EACL-99)*, 1999, p. 110–117.
- THE FREE SOFTWARE FOUNDATION Categories of Free and Non-Free Software. [Http://www.gnu.org/philosophy/categories.html](http://www.gnu.org/philosophy/categories.html); Visitado em Julho de 2006, 2006.
- TURINE, M. A. S. *HMBS: Um Modelo Baseado em Statecharts para a Especificação Formal de Hiperdocumentos*. Tese de Doutorado, Instituto de Física de São Carlos, Universidade de São Paulo, 1998.
- TYRRELL, S. The Many Dimensions of the Software Process. *Crossroads*, v. 6, n. 4, p. 22–26, 2000.
- UWADIA, C. O.; IFINEDO, P. E.; NWAMARAH, G. M.; ESEYIN, E. G.; SAWYERR, A. Risk Factors in the Collaborative Development of Management Information Systems for Nigerian Universities. *Information Technology for Development*, v. 12, n. 2, p. 91–111, 2006.
- VAN DER VEN, J. S.; JANSEN, A. G. J.; NIJHUIS, J. A. G.; BOSCH, J. Design Decisions: The Bridge Between Rationale and Architecture. In: *A. H. Dutoit and R. McCall and I. Mistrík and B. Paech (eds.), Rationale Management in Software Engineering*, Springer, p. 329–348, 2006.

- WALKER, A. Level 5 Process Capability Achievement: A Case Study from Software Engineering Research Management. *Software Process: Improvement and Practice*, v. 8, n. 1, p. 51–62, 2003.
- WALKER, A.; BRAUDE, B. Quality Aspects of Software Product Supply and Support using the Internet. In: *ISESS '99: Proceedings of the 4th IEEE International Symposium and Forum on Software Engineering Standards*, Washington, Estados Unidos: IEEE Computer Society, 1999, p. 199.
- WALKER, C. K. The Need to Provide Writing Support for Academic Engineers. *IEEE Transactions on Professional Communication*, v. 36, n. 3, p. 130–136, 1993.
- WIEGERAAD, S. *Development of a Design History Information System*. Tese de Doutorado, Universiteitsdrukkerij Technische Universiteit Eindhoven, 245 páginas, 1999.
- WILLIAMS, L.; COCKBURN, A. Agile Software Development: Its about Feedback and Change. *IEEE Computer*, v. 36, n. 6, p. 39–43, 2003.
- WOHLIN, C.; RUNESON, P.; HOST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in Software Engineering*. Kluwer Academic Publishers, 2000.
- WULF, W. A. *The National Collaboratory - a White Paper*. Technical report, Appendix A in Towards a National Collaboratory, the unpublished report of an invitational workshop held at the Rockefeller University, 1989.
- YANG, Q. Z.; MIAO, C. Y.; ZHANG, Y.; GAY, R. Ontology Modelling and Engineering for Product Development Process Description and Integration. In: *Proceedings of IEEE International Conference on Industrial Informatics*, Singapura, 2006, p. 85–90.
- YAP, M. Follow the Sun: Distributed Extreme Programming Development. In: *Proceedings of the Agile Development Conference*, Minneapolis, Estados Unidos, 2005, p. 218–224.
- ZHAO, L.; ELBAUM, S. Quality Assurance under the Open Source Development Model. *Journal of Systems and Software*, v. 66, n. 1, p. 65–75, 2003.
- ZORZATTO, J.; DODONOV, E.; QUAINI, J.; GUARDIA, H. C. ProGrid Portal: Integrated Management and Security for Users, Resources and Applications. In: *4th International Information and Telecommunication Technologies Symposium*, Florianópolis - SC, 2005.

Práticas de Desenvolvimento de Software em Projetos de Pesquisa

São apresentados a seguir projetos científicos e as respectivas práticas cumpridas para desenvolvimento. O levantamento apresentado foi realizado com o objetivo de auxiliar na identificação do processo atual executado no desenvolvimento de projetos de pesquisa.

1. **Projeto Genesis:** definição de tema, gerenciamento de conhecimento, análise de requisitos, registro de decisões de plataforma, planejamento de infra-estrutura, planejamento das tarefas para divulgação do projeto, suporte à comunicação, gerenciamento de riscos, gerenciamento de cronograma, gerenciamento de configuração, testes, revisões técnicas, implementação, introdução de elementos de XP (Aversano et al., 2004)
2. **Projeto COSPA:** análise de requisitos (questionários, entrevistas, histórias dos usuários), desenvolvimento de protótipo, análise custo/benefício da aplicação, construção e disponibilização de repositório de conhecimento (Succi e Zuliani, 2004)
3. **Projeto MVCASE:** análise de requisitos, avaliação e melhoria de usabilidade, documentação, gerenciamento de pessoas, projeto arquitetural, implementação, testes, gerenciamento de cronograma, gerenciamento de mudanças (descrição do pedido de mudança, análise do pedido, implementação e disponibilização da mudança) (Almeida et al., 2002; Prado e Lucrédio, 2000, 2001)
4. **Projeto CARA** (*Computer Assisted Resuscitation Algorithm*): especificação formal (usando linguagens para especificação de sistemas de tempo real), análise de requisitos (documento de requisitos gerado usando linguagem natural), projeto

arquitetural, projeto de dados, implementação, testes (plano de testes, casos de testes, execução), avaliação de usabilidade, controle de versões, registro de *design rationale*, inspeções (provas formais), gerenciamento de cronograma, orçamento e pessoas, avaliação e melhoria de usabilidade, revisões técnicas, verificação, treinamento (Guan et al., 2004; Martin, 2004)

5. **Projeto CAOS** (*Collaborative Application for Open Storage*): gerenciamento de pessoas, análise de negócios, análise e projeto de software (análise, documentação e manutenção de casos de uso, requisitos, arquitetura e modelo de base de dados), teste de software (plano de testes, incluindo casos de teste e critérios de aceitação, implementação, execução e validação de casos de teste, fornecimento de relatório de teste), gerenciamento de configuração, planejamento, gerenciamento de riscos, resolução de problemas, treinamento, garantia de qualidade de software, verificação, coleta e registro de métricas, avaliação do projeto (repositorio Java.net <https://caos.dev.java.net/source/browse/caos/RelatoExperiencia/>)
6. **ADVS** (*Lawyer Activities Control System*): análise de requisitos (casos de uso), teste de software (definição de casos de testes, definição de procedimentos de testes), definição de papéis e responsabilidades, gerência de configuração (identificação de configuração, controle de mudança e configuração, controle de versões), treinamento, plano de desenvolvimento do projeto, gerenciamento de riscos (identificação dos principais riscos ao projeto, implementação, procedimentos de análise dos riscos), gerenciamento de custos e de prazos (<http://advs.tigris.org/>)
7. **Projeto Hyperdev**: análise de requisitos (identificação de requisitos gerais, requisitos funcionais, requisitos da interface com o usuário), análise e projeto (diagrama de classes, projeto arquitetural, modelagem da base de dados, definição de módulos e funcionalidades associadas a cada um) desenvolvimento de protótipos evolutivos, documentação, gerenciamento de atividades (coordenação das atividades, determinação da prioridade das tarefas e planejamento das atividades da versão seguinte do sistema), elaboração do manual de uso do software (Maidantchik, 1999)
8. **Projeto Canto Livre**: análise de requisitos (documento de requisitos e casos de uso), projeto arquitetural, testes, planejamento de atividades e acompanhamento do progresso do projeto (por meio de reuniões e do uso da ferramenta Atlassian, que auxiliou o acompanhamento da equipe distribuída), revisões, planejamento do projeto, elaboração de propostas técnicas e comerciais, gerenciamento de alterações e gerenciamento das atividades usando o *Issue Tracker* (do projeto Tigris), controle de versões, implementação, estimativa de esforço (Brito et al., 2004)
9. **Projeto Tupan**: avaliação dos processos e produtos de trabalho, auditoria de processo, gerenciamento de versões, análise de requisitos (documento de requisitos

-
- e casos de uso), análise do projeto, projeto arquitetural, implementação, testes, acompanhamento do progresso do projeto, revisões, planejamento do projeto, gerenciamento de alterações, estimativa de esforço (Spindola et al., 2004)
10. **Protótipo do Modelo APSEE**: análise de requisitos (diagrama de pacotes, diagrama de classes, transição de estados) projeto arquitetural, implementação, identificação de componentes, construção de protótipo, estudos de caso (Reis, 2003a)
 11. **Prosoft Cooperativo**: especificações formais, especificação dinâmica dos processos que implementam a proposta, implementação (Reis, 1998)
 12. **Controle de Acesso para Sistemas Distribuídos Java**: análise de requisitos, projeto arquitetural, desenvolvimento de protótipo (<https://ablp.dev.java.net/>), (Sodre, 2001)
 13. **Projeto Balehar**: análise de requisitos, diagrama de classes, projeto de dados, elaboração do manual de usuário, implementação (<http://balehar.tigris.org/>)
 14. **Projeto Codecrawler**: análise de requisitos, registro de lições aprendidas, projeto arquitetural, controle de versões, implementação (Lanza et al., 2005)
 15. *Development of a Design History Information System*: análise de requisitos, projeto arquitetural, projeto de dados, implementação e instalação do sistema, avaliação (estudos de caso) (Wiegeraad, 1999)
 16. **Seurat**: análise de requisitos, projeto arquitetural, implementação, avaliação (Burge e Brown, 2006)
 17. **Projeto OSCAR**: análise de requisitos, projeto arquitetural, teste de unidade, aplicação de práticas de XP (programação em pares, desenvolvimento orientado a testes, desenvolvimento e liberação de *releases*), gerenciamento de conhecimento, gerenciamento de riscos, gerenciamento de cronograma, revisões técnicas, gerenciamento de configuração, implementação, registro de decisões (Boldyreff et al., 2004)
 18. **Projeto Adelfe**: análise de requisitos, modelagem conceitual, implementação (George et al., 2003)
 19. **AssistPro (Integração de Conhecimento em um Ambiente de Desenvolvimento de Software)**: análise de requisitos, descrição de cenários de utilização, gerenciamento de atividades, projeto arquitetural, prototipação (Falbo, 1998)
 20. **Arquitetura para Gerenciamento de Conhecimentos explícitos sobre Processo de Desenvolvimento de Produto**: análise de requisitos, projeto arquitetural, prototipação, projeto de dados, projeto de interface, descrição de cenário de utilização, avaliação (Amaral, 2001)

21. **AutorE**: análise de requisitos, modelagem conceitual, projeto arquitetural, prototipação (Sante, 2003)
22. **Projeto Mae**: análise de requisitos, projeto arquitetural, prototipação (Roshandel et al., 2004)
23. **Projeto Hipikat**: análise de requisitos, projeto arquitetural, prototipação (Cubranic e Murphy, 2003)
24. **Projeto PDE**: definição de requisitos funcionais e não funcionais, análise do projeto (diagrama de atividades, diagrama de sequência, modelo de classes, diagrama de estados, modelo da interação entre objetos), projeto arquitetural, diagrama de colaboração, implementação (Junior, 2004; Junior et al., 2003)
25. **Desenvolvimento do Ambiente ABCDE-Feature**: análise de requisitos, análise do projeto (diagrama de classes), projeto arquitetural, implementação (Oliveira et al., 2001)
26. **Projeto PERF**: análise de requisitos, projeto arquitetural, implementação (Goes, 2001; Goes e Renaux, 2001)
27. **Especificação e Desenvolvimento da Ferramenta C&L**: controle de versões, análise de requisitos, gerenciamento de atividades e de pessoas, descrição de cenários de utilização (mapa de relacionamentos entre cenários), implementação, testes (plano de testes, casos de testes, execução), gerenciamento de alterações, implementação, avaliação (estudo de caso) (Silva et al., 2003)
28. **Projeto Adequas**: análise de requisitos, projeto arquitetural, implementação, avaliação (estudo de caso) (Oliveira, 2002; Oliveira e Belchior, 2003)
29. **Projeto ProGrid**: análise de requisitos, projeto arquitetural, implementação (Zorzatto et al., 2005)
30. **NSCP** (*Distributed System for the Design of a Network and Service Control Platform*): análise de requisitos, projeto arquitetural, projeto de interface, implementação (Ionescu et al., 2005)
31. **Algorithm for Establishing LSPs for Multicast Communication**: análise de requisitos, projeto arquitetural, implementação, avaliação (Esteves et al., 2005)
32. **Projeto WQoS**: análise de requisitos, análise comportamental, implementação, avaliação (estudos de caso) (Moreira et al., 2005)
33. **Projeto Strathcona**: definição de cenário, implementação, avaliação (Holmes e Murphy, 2005)

-
34. **GQMTTool**: definição de requisitos, análise do projeto (diagrama de classes), projeto arquitetural, implementação, avaliação (Lavazza e Barresi, 2005)
 35. ***Application for the Critical Work of Email and Information Management***: gerenciamento de pessoas e definição de papéis, análise de requisitos (documento de requisitos, registro de estórias), estimativa de tempo para desenvolvimento das estórias, revisões técnicas, planejamento, implementação, testes de aceitação (Bellotti et al., 2002)
 36. ***Jikes Research Virtual Machine Project***: implementação, controle de versões, documentação, revisão de código, gerenciamento de alterações, testes de unidade, testes de regressão, treinamento (Alpern et al., 2005)

Processo Padrão de Desenvolvimento de Projetos de Pesquisa

B.1 Processos fundamentais

Para o contexto do desenvolvimento de projetos de pesquisa em software, foram considerados processos que pudessem ser utilizados para (1) aquisição de recursos que geralmente são utilizados para o desenvolvimento de pesquisas, por exemplo, material de consumo, software e equipamentos; (2) iniciação ou definição de uma proposta geral para o desenvolvimento do projeto; (3) desenvolvimento do projeto, que envolve a execução das atividades de ciclo de vida de desenvolvimento de software; (4) operação do protótipo desenvolvido e suporte operacional e (5) manutenção do protótipo para evolução dos projetos de pesquisa.

B.1.1 Processo de Aquisição

O processo de aquisição é composto pelas atividades de preparação para aquisição e aquisição de recursos.

Preparação para aquisição

Identificador: AQ

Proposta: estabelecer as necessidades e objetivos de aquisição das equipes envolvidas com o desenvolvimento de um projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as necessidades de aquisição sejam conhecidas, a estratégia de aquisição (geralmente definida pela(s) agência(s) que financia(m) o projeto ou pela(s) instituição(ões) na(s) qual(is) é desenvolvido) seja conhecida e usada e o fornecedor seja selecionado.

Tarefas: as tarefas relacionadas à atividade de preparação e realização da aquisição foram identificadas a partir das normas ISO/IEC 12207 e ISO/IEC 15504-5. Os artefatos de saída são dependentes das informações solicitadas pela(s) agência(s) que financia(m) o projeto ou pela(s) instituição(ões) na(s) qual(is) é desenvolvido. Apesar disso, são indicadas sugestões para os artefatos de saída.

AQ.1 Definição dos requisitos para aquisição

- **Objetivos:** estabelecer a necessidade de aquisição dos recursos, definir e validar os requisitos referentes à aquisição, conhecer a estratégia e os critérios de aquisição estabelecidos.
- **Artefatos de entrada:** documentos descrevendo as políticas a serem utilizadas para aquisição de recursos
- **Artefatos de saída:**
 1. Plano de aquisição:
 - 1.1. Necessidade de aquisição
 - 1.2. Requisitos do recurso (requisitos funcionais, requisitos de qualidade, etc)
 - 1.3. Limitações de custo
 - 1.4. Requisitos de prazo de entrega
 - 1.5. Requisitos de suporte
 - 1.6. Requisitos de interfaces
 - 1.7. Requisitos de padrões
 - 1.8. Requisitos relacionados a patentes, *copyright* e licenças.

AQ.2 Seleção de fornecedor

- **Objetivos:** avaliar as capacidades dos fornecedores em cumprir os requisitos definidos, avaliar propostas de possíveis fornecedores e selecionar fornecedor
- **Artefatos de entrada:** plano de aquisição
- **Artefatos de saída:**
 1. Solicitação de orçamento:
 - 1.1. Recurso solicitado
 - 1.2. Requisitos do recurso (requisitos funcionais, de qualidade, de interfaces, de padrões, de patentes, de *copyright* e licenças)
 - 1.3. Prazo de entrega

- 1.4. Suporte oferecido
- 1.5. Requisitos sobre o formato de resposta requerido

2. Proposta do fornecedor:
 - 2.1. Valores de custo para os recursos solicitados
 - 2.2. Demais informações requeridas

3. Análise das propostas:
 - 3.1. Documento analisado
 - 3.2. Responsável pela análise
 - 3.3. Critérios de seleção do fornecedor
 - 3.4. Decisão sobre a seleção do fornecedor
 - 3.5. Razões para seleção do fornecedor

Aquisição de Recursos

Identificador: AR

Proposta: efetivar a aquisição dos recursos necessários para o desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que um acordo seja estabelecido entre o coordenador do projeto e o fornecedor, o fornecedor seja monitorado, os recursos obtidos sejam avaliados e o pedido de aquisição seja concluído.

Tarefas: as tarefas relacionadas à atividade de aquisição de recursos foram identificadas a partir das normas ISO/IEC 12207 e ISO/IEC 15504-5. Assim como ocorreu na atividade AQ (Preparação para aquisição), os artefatos de saída são dependentes das informações solicitadas pela(s) agência(s) que financia(m) o projeto ou pela(s) instituição(ões) na(s) qual(is) é desenvolvido, porém, também são indicadas sugestões para os artefatos de saída.

AR.1 Preparação de contrato

- Objetivos: negociar e aprovar contrato.
- Artefatos de entrada: plano de aquisição, proposta do fornecedor
- Artefatos de saída:
 1. Contrato de aquisição
 - 1.1. Recurso solicitado
 - 1.2. Custo
 - 1.3. Requisitos do recurso (requisitos funcionais, requisitos de qualidade, etc)
 - 1.4. Limitações de custo

- 1.5. Requisitos de prazo de entrega
- 1.6. Requisitos de suporte
- 1.7. Requisitos de interfaces
- 1.8. Requisitos de padrões
- 1.9. Requisitos relacionados a patentes, *copyright* e licenças
- 1.10. Critérios de aceitação do recurso
- 1.11. Outras informações consideradas pertinentes

AR.2 Monitoramento do fornecedor

- Objetivos: estabelecer e manter comunicação entre coordenador ou membros do projeto e fornecedor, revisar aspectos do desenvolvimento com o fornecedor, considerando planos e contrato.
- Artefatos de entrada: plano de aquisição, contrato de aquisição
- Artefatos de saída:
 1. Relatório sobre monitoramento do fornecedor
 - 1.1. Registros sobre o progresso do desenvolvimento
 - 1.2. Registros de desvios e justificativas
 - 1.3. Decisões

AR.3 Aceitação do recurso solicitado

- Objetivo: avaliar os recursos considerando os requisitos e o acordo com o fornecedor.
- Artefatos de entrada: plano de aquisição, contrato de aquisição, recurso fornecido
- Artefatos de saída:
 1. Aceitação de recursos:
 - 1.1. Data de entrega
 - 1.2. Registros de recepção, entrega e aceitação
 - 1.3. Identificação dos recursos entregues
 - 1.4. Aceitação (sim/não) e justificativa
 - 1.5. Identificação do responsável pela aceitação

B.1.2 Processo de Iniciação

O processo de iniciação é composto pelas atividades de escolha do tema, análise de requisitos do sistema, projeto arquitetural do sistema e elaboração de plano de pesquisa.

Escolha do tema

Identificador: ET

Proposta: auxiliar os pesquisadores a escolher um tema de pesquisa de acordo com as aptidões, as tendências e as possibilidades de quem se propõe a elaborar e desenvolver o projeto. Escolher um tema é encontrar um objeto que mereça ser investigado cientificamente e tenha condições de ser formulado e delimitado em função da pesquisa (Marconi e Lakatos, 2002). Conforme observado por Gil (2002), a escolha de um tema que possibilite de fato a realização de uma pesquisa requer bastante habilidade do pesquisador. Assim, o papel do coordenador do projeto é de fundamental importância nesta atividade. Com base em sua experiência ele é capaz de sugerir temas de pesquisa, indicar referências bibliográficas para leitura e advertir quanto às dificuldades que poderão decorrer da escolha de determinados temas.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a comunicação entre os membros envolvidos com a pesquisa seja estabelecida, o tema do projeto e o problema que será resolvido sejam identificados.

Tarefas: as tarefas relacionadas à atividade de escolha do tema foram identificadas a partir de estudos realizados por Gil (2002), Marconi e Lakatos (2001, 2002), Laville e Dionne (1999), Bastos et al. (2004), Frankfort-Nachmias e Nachmias (1992) e Maidantchik (1999).

ET.1 Identificação de temas de interesse

- Objetivo: identificar possíveis temas de pesquisa a partir de estudos bibliográficos, discussões com outros pesquisadores ou percepções de demandas científicas que sejam potenciais objetos de pesquisa.
- Artefatos de entrada: documentos científicos
- Artefatos de saída:
 1. Temas de interesse: registrar situações problema que possam ser objetos de pesquisa do modo mais sucinto possível, porém, com clareza e precisão. Não deve ser negligenciado qualquer parâmetro essencial à compreensão do assunto. Algumas palavras e poucas frases são suficientes. Como sugerido por Laville e Dionne (1999) algumas perguntas devem ser respondidas para cada tema de interesse identificado, como apresentado a seguir.
 - 1.1. Situação-problema identificado
 - 1.2. Motivos pelos quais o problema merece ser pesquisado e benefício(s) que pode(m) ser esperado(s)
 - 1.3. Hipótese a ser explorada (caso já seja possível identificar)
 - 1.4. Informações sobre como o problema pode ser melhor delimitado
 - 1.5. Referências bibliográficas importantes

- 1.6. Questões para discussão com um especialista (caso necessário)
 - 1.7. Conhecimentos que o pesquisador possui que podem ser úteis para ajudá-lo a resolver o problema identificado
 - 1.8. Informações gerais sobre motivação e interesse das pessoas envolvidas (estudante, coordenador do projeto, etc) em relação à resolução do problema identificado
- Informações adicionais, Tabela B.1

Tabela B.1: Informações adicionais relacionadas à tarefa ET.1

Comentários:
No caso de uma pesquisa em equipe, é vantajoso considerar as idéias de cada um dos membros recorrendo à técnica do <i>brainstorming</i> quando for necessário escolher um problema comum de pesquisa. Com a realização de reuniões, deve-se eliminar progressivamente os problemas que parecem menos interessantes e concentrar naqueles que parecem ser os mais interessantes, especificando-os e enriquecendo-os, até que reste apenas um. Ao final, a equipe provavelmente se encontrará diante de um problema de pesquisa de interesse, cuja problemática já estará sendo formulada a partir das discussões (Laville e Dionne, 1999).
Outras informações sobre processos de aquisição de conhecimento, envolvendo técnicas como <i>brainstorming</i> e entrevistas foram apresentadas por Jubileu (1999). Foi proposto um processo de aquisição de conhecimento, denominado IPAIA, contendo cinco fases: início do processo de aquisição de conhecimento, preparação da sessão de aquisição de conhecimento, aquisição do conhecimento, implementação do conhecimento e avaliação do conhecimento.
É fundamental que os valores, as preferências, as experiências de vida, os conhecimentos, os gostos, as curiosidades, as aptidões e as possibilidades das pessoas envolvidas com o desenvolvimento da pesquisa orientem a escolha do tema (Laville e Dionne, 1999; Marconi e Lakatos, 2002).
Referência:
No Capítulo X, Salomon (2004) discute nove principais fontes ou elementos que auxiliam a escolha do tema de pesquisa: observação direta, reflexão, senso comum, experiência pessoal, analogias, observação documental, serendipidade (descoberta repentina e aparentemente causal que leva a uma pesquisa), seminários e controvérsias.

ET.2 Levantamento bibliográfico preliminar

- Objetivo: adquirir conhecimento sobre um tema específico, de interesse dos envolvidos com o desenvolvimento do projeto de pesquisa.
- Artefatos de entrada: temas de interesse

- Artefatos de saída:
 1. Levantamento bibliográfico: para a realização de levantamento bibliográfico podem ser utilizados três procedimentos (Marconi e Lakatos, 2002): pesquisa documental (busca por informações relevantes em arquivos oficiais e particulares, dados históricos e estatísticos, registros em geral e obras literárias), pesquisa bibliográfica (apanhado geral sobre os principais trabalhos já realizados, capazes de fornecer dados atuais e relevantes relacionados ao tema) e contatos diretos (troca de experiências e discussões com pessoas que podem fornecer dados ou sugerir possíveis fontes de informações). Goldenberg (2003) sugere que sejam registrados como resultado do levantamento bibliográfico preliminar:
 - 1.1. Referência completa da obra incluindo autores, título, local e data de publicação
 - 1.2. Objetivos dos autores
 - 1.3. Hipóteses de trabalho
 - 1.4. Metodologia de pesquisa utilizada
 - 1.5. Importância do estudo de acordo com a visão dos autores
 - 1.6. Importância do estudo de acordo com a visão do leitor
 - 1.7. Estudos sugeridos pelos autores, indícios de novas pesquisas identificados
 - 1.8. Resumo da obra
 - 1.9. Avaliação crítica pessoal da obra
 2. Tema de pesquisa: após a realização do levantamento bibliográfico preliminar, o tema a ser explorado no projeto de pesquisa deve ter sido escolhido. A princípio este tema é amplo e oferece apenas uma visão geral do contexto no qual a pesquisa será realizada.
 - 2.1. Tema de pesquisa escolhido
 - 2.2 Justificativa
- Informações adicionais, Tabela B.2

ET.3 Formulação do problema

- Objetivos: especificar o problema a ser tratado no projeto de pesquisa, definir o objetivo do trabalho científico e a motivação para seu desenvolvimento.
- Artefatos de entrada: levantamento bibliográfico, tema de pesquisa
- Artefatos de saída:
 1. Formulação do problema:
 - 1.1. Tema escolhido
 - 1.2. Problema identificado (o problema aqui ainda pode ser amplo)

Tabela B.2: Informações adicionais relacionadas à tarefa ET.2

Sugestões:
Goldenberg (2003) sugere que o registro de temas que não foram escolhidos e a justificativa sejam registrados. Segundo o autor, tal prática é importante porque mantém registrado os motivos pelos quais um tema foi descartado e as restrições que impediram o desenvolvimento de um projeto de pesquisa relacionado ao tema. Tais informações podem ser relevantes para a escolha de tema e definição de problema em projetos futuros.
Recomenda-se que diretrizes para a leitura, análise e interpretação de textos científicos sejam consultadas. Severino (2002), por exemplo, dedica o Capítulo III de seu livro à discussão de elementos relacionados a este assunto.
Comentários:
O levantamento bibliográfico preliminar possibilita que o tema de pesquisa selecionado seja melhor compreendido e que o problema de pesquisa seja definido. Pode ocorrer, por outro lado, que o levantamento bibliográfico contribua para determinar uma mudança nos propósitos iniciais da pesquisa, tendo em vista as dificuldades que o tema pode impor (Gil, 2002).
De acordo com Marconi e Lakatos (2001), a <i>justificativa</i> para desenvolvimento do projeto deve apresentar respostas à questão “por quê?”. Consiste numa exposição sucinta, porém completa, das razões de ordem teórica e dos motivos de ordem prática que tornam importante a realização da pesquisa.

- 1.3. Contexto no qual o problema está inserido
 - 1.4. Abordagens atuais que tratam o problema
 - 1.5. Motivos pelos quais o pesquisador considera que vale a pena resolver este problema ou melhorar as soluções atuais
 - 1.6. Material bibliográfico relevante
- Informações adicionais, Tabela B.3

ET.4 Construção do repositório de terminologias

- Objetivo: descrever os principais termos relativos ao(s) domínio(s) de conhecimento do projeto.
- Artefatos de entrada: documentos científicos, ontologias
- Artefatos de saída:
 1. Repositório de termos acessível a todos os membros da equipe de desenvolvimento. Servirá como um artefato de entrada para desenvolvimento das demais atividades do processo.

Tabela B.3: Informações adicionais relacionadas à tarefa ET.3

Comentário:
De acordo com Gil (2002), não existem regras claras que possam ser aplicadas no processo de formulação do problema. Algumas perguntas, no entanto, podem ser úteis para avaliar em que medida o problema proposto está em condições de ser investigado. Foram sugeridas, por exemplo, as questões: “O tema é de interesse do pesquisador?”; “O problema apresenta relevância teórica e prática?”; “Existe material bibliográfico suficiente e disponível para seu equacionamento e solução?”
Experiência:
Boldyreff et al. (2004) relatam que durante a elaboração do projeto OSCAR foram definidos objetivos de pesquisa difíceis de serem alcançados e avaliados no tempo disponível para desenvolvimento, o que gerou muitos problemas para a equipe responsável. Os autores recomendam muita atenção na fase de elaboração do projeto, de forma que objetivos e resultados factíveis sejam propostos.

Análise de requisitos do sistema

Identificador: AE

Proposta: auxiliar o entendimento e a descrição do contexto geral no qual o problema está inserido e começar a identificar os requisitos técnicos de um sistema relacionado ao tema escolhido e ao problema identificado.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se os requisitos do sistema sejam estabelecidos e sejam priorizados, aprovados e atualizados quando necessário.

Tarefas: as tarefas relacionadas à atividade de análise de requisitos do sistema foram identificadas a partir da norma ISO/IEC 15504.

AE.1 Estabelecer os requisitos do sistema

- Objetivo: estabelecer um conjunto de requisitos do sistema que será tratado no projeto de pesquisa, ou seja, identificar sub-sistemas, requisitos dos usuários e de pesquisadores, elementos de segurança, fatores humanos, elementos de hardware e de equipamentos.
- Artefatos de entrada: formulação do problema, documentos técnicos ou outros artefatos que auxiliem a definição dos requisitos do sistema.
- Artefatos de saída:
 1. Documento de requisitos do sistema
 - 1.1. Visão geral do sistema
 - 1.2. Sub-sistemas identificados
 - 1.3. Requisitos (funções, análise de negócios, requisitos organizacionais,

segurança, ergonomia, restrições do projeto)

1.4. Relacionamento e restrições entre os sub-sistemas

1.5. Elementos de software dos sub-sistemas

1.6. Considerações e restrições para cada sub-sistema em relação a requisitos de memória, de elementos de hardware, de interfaces de usuário, interfaces externas do sistema, desempenho, segurança, proteção de dados, parâmetros, manual de operações e componentes reusáveis

AE.2 Analisar os requisitos do sistema

- Objetivo: analisar os requisitos definidos para o sistema.
- Artefatos de entrada: formulação do problema, documento de requisitos do sistema
- Artefatos de saída:
 1. Documento de análise de requisitos do sistema
 - 1.1. Elementos (ou sub-sistemas) analisados
 - 1.2. Responsável pela análise
 - 1.3. Critério de análise utilizado
 - 1.4. Resultados da análise do sistema
 - 1.5. Aspectos de correitude verificando se os requisitos são completos, compreensíveis, testáveis, consistentes e adequados ao contexto
 - 1.6. Análise da viabilidade dos requisitos identificando quais deles são passíveis de serem considerados no contexto do projeto e no contexto de pesquisa
- Informações adicionais, Tabela B.4

Tabela B.4: Informações adicionais relacionadas à tarefa AE.2

Exemplo:
Na norma ISO/IEC 15504 foram indicadas técnicas que são úteis para cumprimento da atividade de análise de requisitos do sistema: estudo de viabilidade, realização de estudos de caso, prototipação, uso de linguagens formais e realização de <i>workshops</i> .

Projeto arquitetural do sistema

Identificador: PA

Proposta: identificar quais requisitos de sistema devem ser alocados a quais sub-sistemas.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que um projeto da arquitetura do sistema que será tratado no desenvolvimento do projeto de pesquisa seja definido, os requisitos sejam alocados aos sub-sistemas e as interfaces entre os sub-sistemas sejam definidas.

Tarefas: as tarefas relacionadas à atividade de projeto arquitetural do sistema foram identificadas a partir da norma ISO/IEC 15504.

PA.1 Projetar a arquitetura do sistema

- **Objetivos:** estabelecer a arquitetura do sistema em alto nível relacionando componentes de hardware e software, alocar os requisitos do sistema a elementos da arquitetura e definir as interfaces entre os elementos do sistema.
- **Artefatos de entrada:** documento de requisitos do sistema, documento de análise do sistema
- **Artefatos de saída:**
 1. Projeto arquitetural do sistema
 - 1.1. Visão geral do projeto
 - 1.2. Descrição dos elementos do sistema e relacionamento entre eles
 - 1.3. Mapeamento de requisitos para elementos do sistema
 - 1.4. Especificação (projeto) para cada elemento do sistema considerando: requisitos de memória e capacidade, requisitos de interfaces de hardware, requisitos de interface do usuário, requisitos de interface externa, requisitos de desempenho, proteção dos dados, parâmetros, manual de operações, componentes reusáveis.
 - 1.5 Definição de interfaces entre sub-sistemas

Elaboração de plano de pesquisa

Identificador: EP

Proposta: definir um plano que direcione o desenvolvimento do projeto de pesquisa e que apresente um resumo de suas principais características.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que seja elaborado um plano para o desenvolvimento do projeto que sirva como referência para as equipes, podendo ser submetido a uma agência financiadora.

Tarefas: as tarefas relacionadas à atividade de elaboração de plano de pesquisa foram identificadas a partir dos trabalhos de Gil (2002), Barros e Lehfeld (1990), Severino (2002) e Santos (2004).

EP.1 Definição da metodologia científica

- **Objetivo:** definir a metodologia de pesquisa, os procedimentos de pesquisa e as técnicas que serão utilizadas para alcançar os objetivos definidos.
- **Artefatos de entrada:** formulação do problema, documento de atividades do projeto, projeto arquitetural do sistema, documento de requisitos do sistema,

documento de análise de requisitos do sistema, EAP (Estrutura Analítica de Projeto).

- Artefatos de saída:
 1. Documento de descrição da metodologia científica
 - 1.1. Título do projeto
 - 1.2. Hipótese(s)
 - 1.3. Identificação do tipo de pesquisa
 - 1.4. Procedimentos técnicos
 - 1.5. Elaboração dos instrumentos e determinação da estratégia de coleta de dados
 - 1.6. Previsão do plano de análise dos dados
 - 1.7. Previsão da forma de apresentação dos resultados
- Informações adicionais, Tabela B.5

EP.2 Redação do plano de pesquisa

- Objetivo: agrupar as principais características do planejamento do projeto de pesquisa.
- Artefatos de entrada: formulação do problema, documento de atividades do projeto, EAP, documento de definição do escopo, documento de descrição da metodologia científica, lista de atividades do projeto de pesquisa, recursos, cronograma do projeto.
- Artefatos de Saída:
 1. Plano de pesquisa
 - 1.1. Título do projeto
 - 1.2. Formulação do problema
 - 1.3. Hipótese(s)
 - 1.4. Objetivos do projeto
 - 1.5. Justificativa para o desenvolvimento do projeto
 - 1.6. Estado da arte e da prática em relação ao problema
 - 1.7. Metodologia e atividades
 - 1.8. Identificação do tipo de pesquisa
 - 1.9. Procedimentos técnicos
 - 1.10. Elaboração dos instrumentos e determinação da estratégia de coleta de dados
 - 1.11. Previsão do plano de análise dos dados
 - 1.12. Previsão da forma de apresentação dos resultados
 - 1.13. Cronograma de execução da pesquisa

Tabela B.5: Informações adicionais relacionadas à tarefa EP.1

Comentários:
Em relação à identificação do tipo de pesquisa, Gil (1999) ressalta que a classificação mais adotada considera três grupos: <i>pesquisa exploratória</i> que tem como objetivo principal proporcionar uma visão geral de um determinado fato; <i>pesquisa descritiva</i> que tem como objetivo principal descrever as características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis e <i>pesquisa explicativa</i> que tem como objetivo identificar os fatores que determinam ou que contribuem para a ocorrência dos fenômenos.
Após a identificação do tipo de pesquisa é importante selecionar os procedimentos técnicos que serão utilizados. As principais classificações de procedimentos de pesquisa são (Cervo e Bervian, 1983; Gil, 1999; Pádua, 2000): <i>pesquisa bibliográfica</i> , desenvolvida a partir de materiais como livros e artigos científicos; <i>pesquisa documental</i> , realizada a partir de documentos contemporâneos ou retrospectivos, considerados cientificamente autênticos; <i>pesquisa experimental</i> , que consiste na busca de relações entre fatos sociais ou fenômenos físicos pela identificação e pela manipulação das variáveis que determinam a relação causa-efeito; <i>pesquisa de levantamento – survey</i> caracterizada pela interrogação direta das pessoas cujo comportamento deseja-se conhecer, utilizando-se geralmente questionários; <i>estudo de campo</i> em que é estudado um único grupo ou comunidade, buscando-se um aprofundamento maior das questões em estudo; <i>estudo de caso</i> caracterizado pelo estudo profundo e exaustivo de um ou de poucos objetos, de maneira a permitir o seu conhecimento amplo e detalhado; <i>pesquisa-ação</i> realizada com base empírica e associada a uma ação ou à resolução de um problema coletivo, no qual os procedimentos e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo.
As técnicas de coleta de dados constituem um aspecto fundamental no processo de pesquisa. A definição das técnicas a serem utilizadas é dependente dos objetivos da pesquisa e do universo a ser pesquisado. Destacam-se como técnicas para coleta de dados: <i>observação</i> , que é o uso dos sentidos buscando adquirir os conhecimentos necessários para o cotidiano; <i>entrevista</i> , em que o investigador se apresenta frente ao investigado e lhe formula perguntas, com o objetivo de obter dados que interessem à investigação; <i>questionário</i> , que é a técnica de investigação composta por um determinado número de questões apresentadas por escrito às pessoas, tendo por objetivo o conhecimento de opiniões, crenças, sentimentos, interesses, expectativas e situações vivenciadas e <i>documentos</i> , que provém de arquivos históricos, registros, diários, biografias, jornais e revistas (Gil, 1999; Pádua, 2000).

1.14. Definição dos recursos humanos, materiais e financeiros a serem alocados

B.1.3 Processo de Desenvolvimento

O processo de desenvolvimento é composto pelas atividades de definição de funcionalidades do protótipo, planejamento rápido, projeto rápido, desenvolvimento do protótipo, avaliação do protótipo e integração dos resultados das equipes.

Definição de funcionalidades do protótipo

Identificador: DF

Proposta: identificar as funcionalidades que serão implementadas no protótipo.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as funcionalidades do componente software do sistema sejam estabelecidas.

Tarefas: as tarefas relacionadas à atividade de definição de funcionalidades do protótipo foram identificadas a partir da norma ISO/IEC 15504, da norma IEEE 830 (Software Engineering Standards Committee of the IEEE Computer Society, 1998), nos *templates* elaborados no projeto Readysset (*Ready-to-use Software Engineering Templates*)¹ e a partir de Auer e Miller (2002) e Sommerville (2003).

DF.1 Especificar os requisitos do protótipo

- **Objetivo:** documentar as funções do protótipo, os requisitos de usuários, os requisitos organizacionais e os requisitos não funcionais.
- **Artefatos de entrada:** registros de entrevistas com usuários, documentos científicos, documento de análise de requisitos do sistema, projeto arquitetural do sistema,
- **Artefatos de saída:** diferentes notações podem ser utilizadas para a especificação de requisitos do protótipo. Por exemplo, quando métodos ágeis são utilizados, histórias dos usuários são enfatizadas. Outras possibilidades são a especificação de casos de uso e a elaboração de documento de requisitos.

1. Registro de histórias dos usuários

1.1. Título

1.2. Data

1.3. Identificação dos participantes

1.4. História

2. Especificação de casos de uso

2.1. Identificador do caso de uso

2.2. Título do caso de uso

2.3. Autor

¹<http://readysset.tigris.org/nonav/>

- 2.4. Data
 - 2.5. Atores
 - 2.6. Descrição geral do caso de uso
 - 2.7. Fluxo básico de eventos
 - 2.8. Fluxo alternativo de eventos
-
3. Documento de requisitos
 - 3.1. Introdução
 - 3.2. Proposta
 - 3.3. Escopo
 - 3.4. Definições, acrônimos e abreviações
 - 3.5. Referências
 - 3.6. Perspectiva do protótipo (apresenta o protótipo a ser desenvolvido no contexto de outros projetos de pesquisa em desenvolvimento pelo grupo)
 - 3.7. Funções do protótipo
 - 3.8. Características dos usuários
 - 3.9. Restrições
 - 3.10. Suposições e dependências
 - 3.11. Requisitos funcionais
 - 3.12. Requisitos não-funcionais
 - 3.13. Critérios de verificação e validação
- Informações adicionais, Tabela B.6

Planejamento rápido

Identificador: PR

Proposta: definir o planejamento do projeto (para a iteração corrente, caso seja utilizada uma abordagem iterativa).

Resultados: como resultado da implementação bem sucedida desta atividade, obtém-se um planejamento para o desenvolvimento do projeto.

Tarefa: a tarefa relacionada à atividade de planejamento rápido foi identificada a partir dos *templates* elaborados no projeto Readysset (*Ready-to-use Software Engineering Templates*)² e a partir de Auer e Miller (2002).

PR.1 Planejamento

- **Objetivo:** realizar um planejamento elementar que inclui estimativa de tempo para implementar cada estória, caso de uso ou requisito, custos associados,

²<http://readysset.tigris.org/nonav/>

Tabela B.6: Informações adicionais relacionadas à tarefa DF.1

Explicações:
Estórias descrevem como os usuários poderão interagir com o sistema para cumprir uma tarefa. Estórias dos usuários são breves parágrafos (3 a 5 sentenças) que descrevem um cenário específico. Nestas estórias não são considerados detalhes do sistema pois o foco está na interação do usuário com o sistema Auer e Miller (2002).
Um caso de uso é uma descrição de um conjunto de sequência de ações, incluindo variantes realizadas pelo sistema para produzir um resultado observável do ponto de vista de um ator (Booch et al., 2006).
Referências:
Descrições relativas ao conteúdo e à qualidade de um documento de requisitos de software são apresentadas na norma IEEE 830 (Software Engineering Standards Committee of the IEEE Computer Society, 1998). De forma geral, em relação ao conteúdo, devem ser englobadas introdução, descrição geral e especificação de requisitos. Em relação à qualidade, os requisitos apresentados devem ser corretos, não-ambíguos, completos e consistentes. Além disso, prioridades devem ser indicadas e deve ser possível verificá-los, modificá-los e rastreá-los.
É importante notar que outras técnicas além da realização de sessões de <i>brainstormings</i> e revisão da literatura podem ser consideradas na identificação de funcionalidades do protótipo a ser desenvolvido, por exemplo, entrevistas e análise de dados de questionários. Outras informações sobre técnicas para engenharia de requisitos são apresentadas por Kotonya e Sommerville (1998).
Experiências:
Conforme observado por Chirouze et al. (2005), comumente o coordenador do projeto de pesquisa desempenha o papel de usuário final. Neste caso, as estórias podem ser registradas a partir de reuniões individuais entre pesquisador e coordenador do projeto ou a partir de sessões de <i>brainstormings</i> envolvendo as equipes.
Segal (2005) apresenta suas experiências com o desenvolvimento de projetos de pesquisa no laboratório LABtoSPACE e destaca que os cientistas desenvolvem software de forma muito iterativa e os requisitos emergem nas sucessivas iterações. A autora apresenta as dificuldades relacionadas à elaboração e utilização de um documento de requisitos e sugere que seja utilizada uma metodologia ágil e iterativa, que favoreça a inclusão de novos requisitos à medida que o projeto é desenvolvido.

riscos e prioridade (para a iteração corrente, caso seja utilizada uma abordagem iterativa).

- Artefatos de entrada: registro de estórias dos usuários, especificação de casos de uso, documento de requisitos, cronograma de cada equipe
- Artefatos de saída:

1. Planejamento
 - 1.1. Identificador das estórias, requisitos ou casos de uso que serão considerados
 - 1.2. Estimativa de tempo para implementação
 - 1.3. Prioridade (alta, moderada, baixa)
- Informações adicionais, Tabela B.7

Tabela B.7: Informações adicionais relacionadas à tarefa PR.1

Sugestão:
Conforme sugerido por Chirouze et al. (2005), o planejamento deve ser realizado pelas equipes e as decisões tomadas devem ser mantidas em um repositório online, de forma que os membros do projeto possam acessá-las facilmente.

Projeto (*design*) rápido

Identificador: PP

Proposta: definir um projeto simples do protótipo

Resultados: como resultado da implementação bem sucedida desta atividade, obtém-se o projeto do protótipo (para a iteração corrente, caso seja utilizada uma abordagem iterativa).

Tarefas:

PP.1 Elaboração de projeto simples

- Objetivo: elaborar projeto considerando as prioridades estabelecidas no planejamento.
- Artefatos de entrada: registro de estórias dos usuários, diagrama de casos de uso, documento de requisitos, planejamento.
- Artefatos de saída: há diversas possibilidades em relação aos artefatos a serem gerados, por exemplo, diagrama de classes, projeto da arquitetura do sistema, projeto de dados, projeto da interface, diagrama de colaboração e outros sugeridos pela linguagem UML.
- Informações adicionais, Tabela B.8

Desenvolvimento do protótipo

Identificador: DP

Proposta: desenvolver o protótipo e realizar testes de unidade

Resultados: como resultado da implementação bem sucedida desta atividade, o código fonte do protótipo é desenvolvido e testes de unidade são realizados.

Tabela B.8: Informações adicionais relacionadas à tarefa PP.1

Sugestão:
Chirouze et al. (2005) sugeriram, no contexto da metodologia XR (<i>eXtreme Researching</i>), que a atividade de projeto seja baseado em componentização e orientação a objetos, pois, segundo eles, é uma forma de facilitar a transmissão de conhecimento entre as equipes que trabalham em um projeto de pesquisa e facilitar a interação quando o projeto é desenvolvido de forma multidisciplinar.
Referência:
Descrições sobre os diagramas que podem ser gerados na tarefa de elaboração de projeto simples podem ser encontradas em Booch et al. (2006).

Tarefas: as tarefas relacionadas à atividade de desenvolvimento de protótipo foi identificada a partir dos *templates* elaborados no projeto Readysset (*Ready-to-use Software Engineering Templates*)³ e do trabalho de Filho (2001a).

DP.1 Implementação do código fonte

- Objetivo: desenvolver o código fonte do protótipo, considerando as estórias dos usuários, casos de uso ou requisitos elicitados que foram priorizados. Padrões de codificação (*coding standards*) devem ser estabelecidos, divulgados e utilizados.
- Artefatos de entrada: descrição das estórias, casos de uso ou requisitos da iteração corrente, diagramas gerados na atividade de projeto rápido
- Artefatos de saída: código fonte

DP.2 Projetar procedimentos e casos de testes de unidade

- Objetivo: projetar os procedimentos e casos de teste para testes de unidade.
- Artefatos de entrada: planejamento, artefatos produzidos na tarefa de elaboração de projeto simples, código fonte
- Artefatos de saída:
 1. Procedimentos de teste
 - 1.1. Objetivo
 - 1.2. Fluxo
 2. Projeto de casos de teste
 - 2.1. Entradas
 - 2.2. Saídas esperadas

Tabela B.9: Informações adicionais relacionadas à tarefa DP.2

Explicação:
Os procedimentos de teste contêm uma sequência de ações que devem ser executadas para realizar um grupo de testes semelhantes. O fluxo de procedimento representa os passos que devem ser executados por um testador em termos de telas, campos e comandos. Os valores dos campos são determinados nos casos de teste. Portanto, os casos de teste contêm valores de entrada e valores de saída esperados para cada instância de teste (Filho, 2001a).

- Informações adicionais, Tabela B.9

DP.3 Executar testes de unidade

- Objetivo: realizar testes de unidade de acordo com o projeto de casos de teste
- Artefatos de entrada: procedimentos de teste, projeto de casos de teste, código fonte
- Artefatos de saída: código fonte, relatório de testes de acordo com os interesses e necessidades da equipe
- Informações adicionais, Tabela B.10

Avaliação do protótipo

Identificador: AP

Proposta: realizar testes funcionais no protótipo

Resultados: como resultado da implementação bem sucedida desta atividade, o protótipo é avaliado, ou seja, um conjunto de funcionalidades é avaliado pelos usuários finais e um conjunto de alterações é sugerido.

Tarefas: as tarefas relacionadas à atividade de avaliação do protótipo foram identificadas a partir do trabalho de Humphrey (1999) e Filho (2001a).

AP.1 Planejar testes funcionais

- Objetivos: identificar técnica(s) que serão usadas para realização de testes funcionais, especificar condições de término dos testes e determinar recursos para realização de testes.
- Artefatos de entrada: registro de estórias dos usuários, especificação de casos de uso, documento de requisitos, planejamento

³<http://readysset.tigris.org/nonav/>

Tabela B.10: Informações adicionais relacionadas à tarefa DP.3

Sugestões:
Conforme sugerido por Ambati e Kishore (2004), as pessoas que trabalham no desenvolvimento de um módulo ou de uma parte de um projeto de pesquisa são aquelas que conhecem mais sobre os <i>bugs</i> e as restrições que fazem parte do mesmo. Os autores sugerem o cumprimento da atividade de rastreamento de <i>bugs</i> (<i>bug tracking</i>), tendo em vista a alta rotatividade dos membros dos projetos. Outro fator que motiva o cumprimento desta atividade se refere ao fato de que, à medida que os projetos crescem, os problemas causados pelos <i>bugs</i> tendem a se tornar mais complexos. Dessa forma, caso as descrições dos <i>bugs</i> não sejam registradas e disponibilizadas, é provável que as manutenções realizadas pelos desenvolvedores comprometam a qualidade e os objetivos a longo prazo do projeto.
A ferramenta Bugzilla pode ser útil para contribuir na implantação da atividade de rastreamento de <i>bugs</i> . Em especial, o uso da ferramenta pode colaborar para que a comunicação ocorra quando o projeto é desenvolvido de forma distribuída e para registrar informações sobre <i>bugs</i> que serão úteis quando novos membros assumirem o projeto. A descrição das principais funcionalidades da ferramenta Bugzilla pode ser encontrada em Quemello et al. (2005).

- Artefatos de saída:
 1. Planejamento de testes funcionais
 - 1.1. Técnica(s) de teste a ser(em) utilizada(s)
 - 1.2. Itens a serem testados
 - 1.3. Grau de cobertura por itens
 - 1.4. Condições de término normal dos testes
 - 1.5. Condições de término anormal dos testes e procedimentos a adotar
 - 1.6. Responsabilidades, cronograma e recursos necessários
- Informações adicionais, Tabela B.11

Tabela B.11: Informações adicionais relacionadas à tarefa AP.1

Comentário:
Avaliações com clientes e usuários são práticas fortemente recomendadas no contexto de metodologias ágeis. Em ambiente de desenvolvimento de pesquisa, a realização de avaliações pode ser favorecida, pois o cliente ou usuário (às vezes o próprio coordenador do projeto) está presente no ambiente de desenvolvimento (<i>on site customer</i>) (Chirouze et al., 2005).

AP.2 Projetar testes funcionais

- Objetivo: projetar os procedimentos e casos de teste.
- Artefatos de entrada: registro de estórias dos usuários, especificação de casos de uso, documento de requisitos,
- Artefatos de saída:
 1. Projeto dos procedimentos de teste
 - 1.1. Objetivo
 - 1.2. Fluxo
 2. Projeto de casos de teste
 - 2.1. Entradas
 - 2.2. Saídas esperadas

AP.3 Executar testes funcionais

- Objetivo: realizar testes de acordo com o planejamento de testes funcionais, projeto dos procedimentos de testes funcionais e projeto de casos de teste funcionais.
- Artefatos de entrada: planejamento dos testes funcionais, procedimentos e casos de teste, registro de estórias dos usuários, especificação de casos de uso, documento de requisitos, código fonte
- Artefato de saída:
 1. Relatório de testes funcionais:
 - 1.1. Data (ou período) em que os testes foram realizados
 - 1.2. Identificação do responsável(is) pela execução dos testes
 - 1.3. Protótipo e configuração testados
 - 1.4. Tempo gasto na execução dos testes
 - 1.5. Descrição dos defeitos encontrados e medidas adotadas
 - 1.6. Outros resultados dos testes relevantes

Integração dos resultados das equipes

Identificador IR

Proposta: realizar a integração dos resultados parciais obtidos com o desenvolvimento do projeto.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que os protótipos desenvolvidos pelas equipes sejam integrados e testados.

Tarefas: as tarefas relacionadas à atividade de integração dos resultados das equipes foram identificadas a partir do trabalho de Humphrey (1999), Paasvaara e Lassenius (2004) e Filho (2001a).

IR.1 Planejar a integração dos resultados de cada equipe

- Objetivos: elaborar um plano de integração dos artefatos produzidos pelas equipes e realizar a integração.
- Artefatos de entrada: artefatos produzidos por cada equipe e que serão integrados, projeto arquitetural do protótipo
- Artefatos de saída:
 1. Plano de integração:
 - 1.1. Módulos (componentes) e artefatos a serem integrados
 - 1.2. Estratégia de teste de integração a ser utilizada (*top-down ou bottom-up*)
 - 1.4. Responsabilidades, cronograma e recursos necessários
- Informações adicionais, Tabela B.12

Tabela B.12: Informações adicionais relacionadas à tarefa IR.1

Comentários:
Conforme apresentado por Paasvaara e Lassenius (2004), muitos benefícios podem ser alcançados com a realização de integração após a execução de cada ciclo de iteração. Por exemplo, previne-se que as equipes desenvolvam os projetos por longos períodos independentemente e produzam módulos difíceis de integrar e, além disso, o trabalho das equipes torna-se mais transparente.
Sempre que um novo módulo é integrado ao sistema, podem ocorrer problemas com funções que anteriormente eram executadas de forma correta. Portanto, devem ser realizados testes de regressão que é a re-execução de um conjunto de testes para assegurar que as mudanças não incluíram erros adicionais. O subconjunto de testes a ser executado contém, principalmente, uma amostra representativa de testes que irão exercitar todas as funções de software e testes adicionais com foco em funções que são prováveis de serem afetadas pelas mudanças e testes (Pressman, 2005).

IR.2 Projetar testes de integração

- Objetivo: projetar os procedimentos e casos de teste de integração.
- Artefatos de entrada: projeto arquitetural do software
- Artefatos de saída:
 1. Projeto dos procedimentos de teste
 - 1.1. Objetivo
 - 1.2. Fluxo
 2. Projeto dos casos de teste
 - 2.1. Entradas

2.2. Saídas esperadas

IR.3 Executar testes de integração

- Objetivos: fazer a integração dos artefatos produzidos pelas equipes e realizar testes de acordo com o planejamento de testes de integração, procedimentos de testes de integração e projeto de casos de teste de integração.
- Artefatos de entrada: projeto arquitetural do software, plano de integração, procedimentos e casos de teste de integração, artefatos integrados
- Artefatos de saída:
 1. Relatório de teste de integração
 - 1.1. Data (ou período) em que os testes foram realizados
 - 1.2. Identificação do responsável(is) pela execução dos testes
 - 1.3. Protótipo e configuração testados
 - 1.4. Tempo gasto na execução dos testes
 - 1.5. Descrição dos defeitos encontrados e medidas adotadas
 - 1.6. Outros resultados relevantes dos testes

B.1.4 Processo de Operação

O processo de operação é composto pelas atividades de garantia do uso operacional do protótipo e suporte ao usuário.

Garantia do uso operacional do protótipo

Identificador: GU

Proposta: assegurar a operação correta e eficiente do protótipo de acordo com a intenção de uso dos usuários em seu ambiente de execução.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que testes operacionais do protótipo sejam realizados e os usuários sejam orientados em relação à operação do protótipo.

Tarefa: as tarefas relacionadas à atividade de garantia do uso operacional do protótipo foram identificadas a partir da norma ISO/IEC 15504.

GU.1 Executar testes operacionais

- Objetivo: executar teste operacional de cada *release* do protótipo, avaliando a satisfação do usuário quando o protótipo é executado em seu ambiente de operação.

- Artefatos de entrada: registro de estórias dos usuários, especificação de casos de uso, documento de requisitos, protótipo
- Artefatos de saída: deve ser gerado um relatório apresentando resultados de medidas de interesse, de acordo com o tipo de aplicação, que indiquem se o protótipo opera satisfatoriamente no ambiente de execução real. Algumas sugestões para o relatório são apresentadas a seguir:
 1. Relatório de testes operacionais
 - 1.1. Recursos de memória necessários para execução
 - 1.2. Tempo de execução
 - 1.3. Desempenho operacional
 - 1.4. Restrições observadas
 - 1.5. Decisões

Suporte ao usuário

Identificador: SU

Proposta: estabelecer e manter um nível aceitável de serviços, prestando assistência e consultoria ao usuário para que o uso do protótipo seja efetivo.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as necessidades de suporte ao usuário sejam identificadas, monitoradas e atendidas

Tarefa: as tarefas relacionadas à atividade de suporte ao usuário foram identificadas a partir da norma ISO/IEC 15504.

SU.1 Estabelecer e implantar um mecanismo de suporte

- Objetivo: estabelecer um serviço por meio do qual o usuário possa registrar e comunicar problemas e dúvidas encontradas durante o uso do protótipo e receber ajuda na resolução. Ferramentas e infra-estrutura para suporte devem ser implantadas quando necessário.
- Artefatos de entrada: protótipo, recursos de auxílio ao suporte
- Artefatos de saída:
 1. Suporte
 - 1.1. Identificação do responsável pelo pedido de suporte
 - 1.2. Descrição do problema
 - 1.3. Classificação do problema (crítico, urgente, relevante, etc)
 - 1.4. *Status* do problema
 - 1.5. Ações e resultados

B.1.5 Processo de Manutenção

O processo de manutenção é composto pelas atividades de preparação para manutenção e realização de manutenção no protótipo

Preparação para manutenção

Identificador: PM

Proposta: planejar as atividades que serão cumpridas para a realização da manutenção do protótipo.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que o processo a ser cumprido para a manutenção seja definido.

Tarefa: a tarefa relacionada à atividade de preparação para manutenção foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

PM.1 Desenvolver estratégia de manutenção

- Objetivo: desenvolver estratégia para gerenciar modificações
- Artefatos de entrada: políticas para realização de manutenção
- Artefatos de saída:
 1. Plano para manutenção
 - 1.1. Procedimentos para receber, registrar e rastrear registros de problemas e pedidos de modificação dos usuários e envio de resposta.
 - 1.2. Procedimentos para cumprimento da atividade de manutenção

Realizar manutenção no protótipo

Identificador: RM

Proposta: cumprir atividades de manutenção de acordo com a estratégia de manutenção definida

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a manutenção seja realizada quando pertinente.

Tarefas: as tarefas relacionadas à atividade de realização de manutenção no protótipo foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

RM.1 Analisar o pedido de manutenção

- Objetivo: analisar o registro do problema e o pedido de modificação em relação ao impacto na organização, nos artefatos e nos sistemas que fazem interface com o protótipo.

- Artefatos de entrada: plano para manutenção, solicitação de manutenção, artefatos gerados durante o cumprimento das atividades de definição de funcionalidades do protótipo, projeto rápido e desenvolvimento do protótipo
- Artefatos de saída:
 1. Pedido de mudança
 - 1.1. Descrição do pedido de mudança
 - 1.2. Identificador do pedido de mudança
 - 1.3. *Status* do pedido (novo, aceito, rejeitado, etc)
 - 1.4. Informações de contato do solicitante
 - 1.5. Impacto causado pela modificação
 - 1.6. Data de entrega solicitada pelo usuário

RM.2 Documentar, implementar e testar a modificação

- Objetivos: determinar quais artefatos precisam ser modificados, implementar, testar e documentar as modificações realizadas.
- Artefatos de entrada: plano para manutenção, pedido de mudança, artefatos gerados durante o cumprimento das atividades de definição de funcionalidades do protótipo, projeto rápido e desenvolvimento do protótipo
- Artefatos de saída:
 1. Documentos atualizados
 2. Protótipo modificado e testado
 3. Relatório de testes atualizado
- Informações adicionais, Tabela B.13

Tabela B.13: Informações adicionais relacionadas à tarefa RM.2

Comentário:
Na norma ISO/IEC 12207 é destacada a importância em cumprir as atividades do processo de desenvolvimento quando uma manutenção é solicitada

RM.3 Revisar e aceitar modificação

- Objetivo: revisar a modificação realizada e obter aceitação, de forma que a integridade do protótipo seja comprovada.
- Artefatos de entrada: pedido de mudança, documentos atualizados, protótipo modificado e testado e relatório de testes atualizado

- Artefatos de saída:
 1. Relatório de revisão
 - 1.1. Modificação realizada
 - 1.2. Atividades de revisão
 - 1.3. Responsáveis pela revisão
 - 1.4. Resultados da revisão
 - 1.5. Aceitação ou rejeição da modificação

RM.4 Descontinuar o protótipo

- Objetivo: documentar a finalização do suporte à operação e manutenção do protótipo
- Artefatos de entrada: não há
- Artefato de saída:
 1. Encerramento de suporte à operação e manutenção do protótipo:
 - 1.1. Indicação sobre encerramento parcial ou total de suporte
 - 1.2. Forma de arquivamento do protótipo e de sua documentação
 - 1.3. Responsabilidades por necessidades de suporte residuais
 - 1.4. Transição para o novo protótipo, se aplicável
 - 1.5. Acessibilidade a dados
 - 1.6. Justificativa para finalização do suporte
 - 1.7. Data em que o protótipo será descontinuado

B.2 Processos de apoio

De acordo com a norma ISO/IEC 12207, os processos de apoio auxiliam e contribuem para o sucesso e a qualidade do projeto de software.

Para o contexto do desenvolvimento de projetos de pesquisa, foram incorporados processos para (1) estabelecimento dos padrões de documentação que serão utilizados para desenvolvimento do projeto; (2) gerência de configuração; (3) garantia de qualidade de produtos e processos; (4) verificação; (5) validação; (6) revisão; (7) resolução de problemas; (8) revisão sistemática; (9) preparação de documentos científicos; (10) elaboração de módulos educacionais; (11) *postmortem* e (12) transferência tecnológica.

B.2.1 Processo de Documentação

O processo de documentação é composto pelas atividades de implantação do processo, planejamento da documentação e produção da documentação.

Implantação do processo

Identificador: IP

Proposta: determinar a estratégia de produção da documentação, tratando o que deve ser documentado por cada equipe em cada um dos estágios do ciclo de vida. Deve-se estabelecer também onde os documentos serão armazenados.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma política para produção da documentação seja definida.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir das normas ISO/IEC 15504-5 e ISO/IEC 12207 e do trabalho de Maidantchik (1999).

IP.1 Desenvolver política para documentação

- Objetivo: estabelecer padrões para desenvolver, modificar e manter documentos.
- Artefato de entrada: documentos indicando exigências em relação à documentação
- Artefato de saída:
 1. Plano para documentação
 - 1.1. Título do documento
 - 1.2. Objetivo
 - 1.3. Audiência
 - 1.4. Procedimentos e responsabilidades para a produção, revisão, modificação, aprovação, produção, armazenamento, manutenção e gerência de configuração dos documentos

Planejamento da documentação

Identificador: PD

Proposta: definir os padrões de documentação que deverão ser usados pelas equipes.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que seja definido como a documentação deverá ser elaborada pelos participantes do desenvolvimento do projeto de pesquisa.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir da norma ISO/IEC 15504-5 e ISO/IEC 12207 e do trabalho de Maidantchik (1999).

PD.1 Gerar modelos para documentação

- Objetivo: definir o padrão a ser utilizado na elaboração de cada documento e indicar as fontes para obtenção de informações.

- Artefato de entrada: normas e documentos científicos descrevendo padrões de documentação
- Artefato de saída:
 1. Modelo para documentação:
 - 1.1. Informações sobre formato do documento
 - 1.2. Informações sobre descrição do conteúdo
 - 1.3. Informações sobre apresentação do documento

Produção da documentação

Identificador: PO

Proposta: produzir a documentação de acordo com o plano e o modelo para documentação elaborados.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a documentação seja elaborada e satisfaça ao plano e ao modelo para documentação estabelecidos.

Tarefas: as tarefas relacionadas à atividade de produção da documentação foram identificadas a partir das normas ISO/IEC 15504-5 e ISO/IEC 12207 e do trabalho de Maidantchik (1999).

PO.1 Documentar o desenvolvimento do projeto de pesquisa

- Objetivo: produzir a documentação do projeto que está sendo desenvolvido. Os documentos devem poder ser acessados pelas equipes utilizando-se rede de computadores. Manutenções da documentação, quando necessárias, devem ser realizadas de acordo com o processo de manutenção e gerência de configuração.
- Artefatos de entrada: plano para documentação, modelo para documentação
- Artefato de saída:
 1. Documentos produzidos

B.2.2 Processo de Gerenciamento de Configuração

O processo de gerenciamento de configuração é composto pelas atividades de implantação do processo, identificação de itens de configuração, controle da configuração, avaliação da configuração.

Implantação do processo

Identificador: IO

Proposta: propor uma política para gerenciamento de configuração que deverá ser utilizada pelas equipes que participam do desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma política de gerenciamento de configuração seja definida e implantada.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207 e do trabalho de Maidantchik (1999).

IO.1 Desenvolver política para gerenciamento de configuração

- Objetivo: determinar as atividades de gerenciamento de configuração que serão cumpridas pelas equipes
- Artefato de entrada: documentos indicando exigências em relação ao gerenciamento de configuração
- Artefato de saída:
 1. Plano para gerenciamento de configuração
 - 1.1. Procedimentos para controlar mudanças nos itens sob gerenciamento de configuração
 - 1.2. Equipes de trabalho responsáveis pelos itens de configuração
 - 1.3. Localização dos itens sob gerenciamento de configuração e mecanismos de acesso
 - 1.4. Mecanismos a serem utilizados para armazenamento e recuperação de itens sob gerenciamento de configuração
 - 1.5. Mecanismos a serem utilizados para gerenciamento de *branches*, estratégias para realização de *merging*, versionamento de arquivos em uma *branch*
 - 1.6. Estabelecimento de *baselines*
 - 1.7. Políticas de criação, alteração e exclusão de arquivos e diretórios

Identificação de itens de configuração

Identificador: II

Proposta: estabelecer os itens que serão gerenciados de acordo com o processo de gerenciamento de configuração

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que sejam estabelecidos os itens cuja configuração será gerenciada.

Tarefa: a tarefa relacionada à atividade de identificação de itens de configuração foi identificada a partir da norma ISO/IEC 15504.

II.1 Estabelecer itens de configuração

- Objetivo: definir os itens de software cuja configuração será gerenciada durante o desenvolvimento do projeto de pesquisa. Devem ser englobados

os itens desenvolvidos por cada equipe e os itens resultantes da integração dos sub-projetos.

- Artefatos de entrada: artefatos do processo e do projeto
- Artefato de saída:
 1. Itens sob gerenciamento de configuração:
 - 1.1. Item sob gerenciamento de configuração
 - 1.2. Tipo do item (por exemplo, modelo de *template*, código fonte, documento)
 - 1.3. Outras informações sobre o item de configuração

Controle da configuração

Identificador: CC

Proposta: realizar efetivamente o gerenciamento de configuração

Resultados: como resultado da implementação bem sucedida deste processo espera-se que o gerenciamento de configuração seja realizado e *baselines* possam ser recuperadas com sucesso quando necessário.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir da norma ISO/IEC 15504.

CC.1 Controlar alterações

- Objetivo: promover o registro e o rastreamento do *status* dos pedidos de alteração e registro de erros nos projetos de pesquisa.
- Artefatos de entrada: itens sob gerenciamento de configuração, itens de configuração, pedido de mudança
- Artefatos de saída:
 1. Pedido de alteração:
 - 1.1. Identificador do pedido de mudança
 - 1.2. avaliação do pedido de mudança e decisão

CC.2 Controlar versões

- Objetivo: combinar procedimentos e ferramentas para gerenciar versões de itens que são criados durante o desenvolvimento do projeto de pesquisa e que estão sob gerenciamento de configuração.
- Artefatos de entrada: repositório de versões, itens sob gerenciamento de configuração, itens de configuração
- Artefatos de saída: o controle de versões envolve a utilização de um repositório de versões e o armazenamento e o gerenciamento de todas as versões dos itens de configuração.

Avaliação da configuração

Identificador: AC

Proposta: garantir que informações sobre itens que estão sob gerenciamento de configuração são completas e consistentes.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que os itens de configuração sejam avaliados e informações de interesse sejam disponibilizadas.

Tarefa: a tarefa relacionada à atividade de avaliação da configuração foi identificada a partir da norma ISO/IEC 15504.

AC.1 Avaliação do aspecto funcional de item de configuração

- **Objetivo:** verificar se foram realizadas as mudanças requisitadas, se a nova versão respeita os requisitos e se o projeto e a documentação estão atualizados.
- **Artefatos de entrada:** pedido de mudança, pedido de alteração, itens de configuração, repositórios
- **Artefatos de saída:**
 1. Relatório de avaliação sobre itens de configuração
 - 1.1. Item de configuração avaliado
 - 1.2. Informações sobre o item de configuração (atendimento à solicitação de mudança, atualização da documentação, etc)

B.2.3 Processo de Garantia da Qualidade

O processo de garantia da qualidade é composto pelas atividades de implantação do processo, garantia de qualidade de artefatos e garantia de qualidade de processos.

Implantação do Processo

Identificador: IC

Proposta: desenvolver uma estratégia para garantia de qualidade dos artefatos e do processo.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para garantia de qualidade seja definida e implantada.

Tarefa: a tarefa relacionada à atividade de implantação do processo foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

IC.1 Desenvolver política para garantia de qualidade

- **Objetivo:** determinar as atividades de garantia de qualidade que serão cumpridas pelas equipes

- Artefato de entrada: documentos indicando exigências em relação à garantia de qualidade
- Artefato de saída:
 1. Plano para garantia de qualidade
 - 1.1. Padrões, métodos, procedimentos e ferramentas para execução de atividades de garantia de qualidade
 - 1.2. Procedimentos para identificação, coleta, registro, manutenção e disponibilização dos registros de qualidade
 - 1.3. Recursos, cronogramas e responsabilidades para o cumprimento de atividades de garantia de qualidade

Garantia de qualidade de artefatos

Identificador: GQ

Proposta: garantir que todos os artefatos produzidos seguem os planos e satisfazem aos requisitos

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que avaliações sobre a conformidade dos artefatos aos padrões aplicáveis sejam realizadas.

Tarefas: as tarefas relacionadas à atividade de garantia de qualidade de artefatos foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

GQ.1 Garantir qualidade dos artefatos

- Objetivos: assegurar que os artefatos são documentados de acordo com os planos, são consistentes e estão sendo elaborados da forma como foi requisitado.
- Artefatos de entrada: plano para garantia de qualidade, artefatos produzidos, métricas, contratos e acordos
- Artefato de saída:
 1. Relatório de garantia de qualidade de artefatos
 - 1.1. Artefato avaliado
 - 1.2. Resultados de medições e observações (pontos favoráveis/restrições) referentes à documentação do artefato
 - 1.3. Resultados de medições e observações (pontos favoráveis/restrições) referentes à consistência do artefato

Garantia de qualidade de processos

Identificador: GA

Proposta: garantir que os processos são cumpridos de acordo com os planos estabelecidos

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que avaliações sobre a conformidade dos processos aos padrões aplicáveis sejam realizadas.

Tarefa: a tarefa relacionada à atividade de implantação do processo foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

GA.1 Garantir qualidade do processo

- **Objetivos:** assegurar que os processos são cumpridos conforme os padrões estabelecidos e satisfazem as exigências de acordos estabelecidos ou de contratos.
- **Artefato de entrada:** plano para garantia de qualidade, padrões de referência, métricas, contratos e acordos
- **Artefato de saída:**
 1. Garantia de qualidade de processos
 - 1.1. Processo avaliado
 - 1.2. Resultados de medições e observações (pontos favoráveis/restrições) sobre garantia de qualidade em relação ao processo

B.2.4 Processo de Verificação

O processo de verificação é composto pelas atividades de implantação do processo e verificação.

Implantação do Processo

Identificador: IE

Proposta: desenvolver uma estratégia para verificação.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para verificação seja definida e implantada.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

IE.1 Desenvolver política para verificação

- **Objetivos:** determinar as atividades de verificação que serão executadas pelas equipes
- **Artefato de entrada:** documentos indicando exigências em relação à verificação
- **Artefatos de saída:**
 1. Plano para verificação
 - 1.1. Métodos, técnicas e ferramentas que serão usadas para verificação
 - 1.2. Artefatos que deverão ser verificados
 - 1.3. Cronograma para execução das atividades de verificação
 - 1.4. Critérios que deverão ser considerados na verificação

Verificação

Identificador: VE

Proposta: verificar se os artefatos estão sendo desenvolvidos da forma certa

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a verificação dos artefatos seja realizada de acordo com o plano para verificação.

Tarefas: as tarefas relacionadas à atividade de verificação foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

VE.1 Verificação dos artefatos

- **Objetivos:** verificar se os artefatos estão corretos e se foram completamente e corretamente integrados.
- **Artefatos de entrada:** plano para verificação, artefatos que serão integrados
- **Artefatos de saída:**
 1. Relatório de verificação
 - 1.1. Item verificado e critérios utilizados
 - 1.2. Itens cuja verificação foi bem sucedida
 - 1.3. Itens que falharam
 - 1.4. Problemas identificados
 - 1.5. Ações recomendadas
 - 1.6. Conclusões da verificação

B.2.5 Processo de Validação

O processo de validação é composto pelas atividades de implantação do processo e validação.

Implantação do Processo

Identificador: IS

Proposta: desenvolver uma estratégia para validação.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para validação seja definida e implantada.

Tarefa: a tarefa relacionada à atividade de implantação do processo foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

IS.1 Desenvolver política para validação

- **Objetivo:** determinar as atividades de validação que serão cumpridas pelas equipes

- Artefato de entrada: documentos indicando exigências em relação à validação
- Artefatos de saída:
 1. Plano para validação
 - 1.1. Métodos, técnicas e ferramentas que serão usadas para validação
 - 1.2. Artefatos que deverão ser validados
 - 1.3. Cronograma para execução das atividades de validação
 - 1.4. Critérios de validação que deverão ser considerados

Validação

Identificador: VA

Proposta: avaliar se os requisitos para a elaboração dos artefatos são satisfeitos.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a validação dos artefatos seja realizada de acordo com o plano para validação.

Tarefa: a tarefa relacionada à atividade de validação foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

VA.1 Validação dos artefatos

- Objetivo: avaliar se os artefatos produzidos estão de acordo com os requisitos e os planos estabelecidos.
- Artefatos de entrada: plano para validação, artefatos que serão validados
- Artefatos de saída:
 1. Relatório de validação
 - 1.1. Item validado e critérios utilizados
 - 1.2. Itens cuja validação foi bem sucedida
 - 1.3. Itens que falharam
 - 1.4. Problemas identificados
 - 1.5. Ações recomendadas
 - 1.6. Conclusões da validação

B.2.6 Processo de Revisão

O processo de revisão é composto pelas atividades de implantação do processo e realização de revisões.

Implantação do Processo

Identificador: IB

Proposta: desenvolver uma estratégia para realização de revisões

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para a realização de revisões seja definida e implantada.

Tarefa: a tarefa relacionada à atividade de implantação do processo foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207 e do trabalho de Maidantchik (1999).

IB.1 Desenvolver política para revisão

- Objetivo: determinar as atividades de revisão que serão cumpridas pelas equipes
- Artefatos de entrada: documentos indicando exigências em relação à revisão
- Artefatos de saída:
 1. Plano para revisão
 - 1.1. Itens a serem revisados
 - 1.2. Recursos necessários para realizar a revisão
 - 1.3. Critérios para revisão (*checklists*, requisitos, padrões)
 - 1.4. Cronograma para revisão
 - 1.5. Mecanismos para preparação do time de revisão
 - 1.6. Procedimentos para realização da revisão
 - 1.7. Artefatos de entrada e saída da revisão
 - 1.8. Experiência esperada de cada revisor
 - 1.9. Registros de revisão a serem armazenados
 - 1.10. Medidas de revisão a serem mantidas

Realização de revisões

Identificador: RR

Proposta: realizar revisões durante o desenvolvimento do projeto de pesquisa em relação ao gerenciamento, à correção técnica dos artefatos produzidos e à qualidade científica dos documentos produzidos.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que revisões sejam realizadas de acordo com o plano para revisão.

Tarefas: as tarefas relacionadas à atividade de realização de revisões foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

RR.1 Realização de revisões gerenciais

- Objetivos: revisar o *status* do projeto de acordo com os planos de revisão estabelecidos.
- Artefatos de entrada: plano para revisão, artefatos que serão revisados

- Artefato de saída:
 1. Relatório de revisão gerencial
 - 1.1. Contexto geral da revisão (o que foi revisado, *status* da revisão)
 - 1.2. *Checklists* usados e critérios de revisão
 - 1.3. Preparação para revisão
 - 1.4. Tempo gasto na revisão
 - 1.5. Revisores, papéis e experiências
 - 1.6. Resultados da revisão sobre o progresso das atividade e o cumprimento dos planos
 - 1.7. Resultados da revisão sobre alocação de recursos e cumprimento dos planos
 - 1.8. Determinação da necessidade de alterar planos

RR.2 Realização de revisões técnicas

- Objetivos: revisar os artefatos de acordo com os planos de revisão estabelecidos.
- Artefatos de entrada: plano para revisão, artefatos que serão revisados
- Artefato de saída:
 1. Relatório de revisão técnica
 - 1.1. Contexto geral da revisão (o que foi revisado, *status* da revisão)
 - 1.2. *Checklists* usados e critérios de revisão
 - 1.3. Preparação para revisão
 - 1.4. Tempo gasto na revisão
 - 1.5. Revisores, papéis e experiências
 - 1.6. Completude dos artefatos gerados
 - 1.7. Revisão sobre atendimento dos artefatos a padrões e especificações
 - 1.8. Aceitação/rejeição do artefato para execução da próxima atividade

RR.3 Realização de revisões de documentos científicos

- Objetivos: revisar documentos que relatam os resultados obtidos com o desenvolvimento de projetos de pesquisa, tais como artigos, dissertações e teses.
- Artefato de entrada: documento científico, critérios para revisão
- Artefatos de saída:
 1. Relatório de revisão de documentos científicos
 - 1.1. Contexto geral da revisão (o que foi revisado, *status* da revisão)
 - 1.2. *Checklists* usados e critérios de revisão
 - 1.3. Parecer final da revisão
- Informações adicionais, Tabela B.14

Tabela B.14: Informações adicionais relacionadas à tarefa RR.3

Referência:
Smith (1990) indica uma série de sugestões para a revisão de artigos científicos e para geração do relatório de revisão. Segundo o autor, é fundamental avaliar um artigo sem qualquer noção pré-concebida de qualidade ou corretitude. A tarefa do revisor é decidir se o conteúdo do documento traz contribuição para o campo de pesquisa para o qual é destinado. Segundo o autor, durante a revisão de um artigo científico devem ser respondidas as seguintes questões: (a) Qual é o propósito do artigo? O problema, os aspetos relevantes e os resultados estão claramente colocados? (b) O artigo é apropriado para o fórum escolhido? (c) O objetivo é relevante? (d) O método aplicado é válido? (e) A execução do método está correta? (f) Conclusões corretas foram abstraídas? (g) a apresentação é satisfatória?

B.2.7 Processo de Resolução de Problemas

O processo de resolução de problemas é composto pelas atividades de implantação do processo e resolução de problemas.

Implantação do processo

Identificador: ID

Proposta: desenvolver uma estratégia para resolução de problemas.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma política para resolução de problemas seja definida e implantada.

Tarefa: a tarefa relacionada à atividade de implantação do processo foi identificada a partir da norma ISO/IEC 12207 e do trabalho de Maidantchik (1999).

ID.1 Desenvolver política de resolução de problemas

- Objetivo: identificar como os problemas detectados durante o desenvolvimento do projeto de pesquisa devem ser tratados.
- Artefatos de entrada: documentos indicando exigências em relação à resolução de problemas
- Artefatos de saída:
 1. Plano para resolução de problemas:
 - 1.1. Mecanismo utilizado para registro de problemas
 - 1.2. Esquema para categorização e priorização de problemas
 - 1.3. Análises que serão realizadas para detectar tendências nos problemas reportados
 - 1.4. Avaliações que serão realizadas quando os problemas forem resolvidos

Resolução de problemas

Identificador: RP

Proposta: assegurar que todos os problemas detectados durante o desenvolvimento do projeto de pesquisa serão identificados, analisados, gerenciados e controlados.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a resolução dos problemas seja realizada de acordo com o plano para resolução de problemas.

Tarefas: as tarefas relacionadas à atividade de resolução de problemas foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

RP.1 Registro do problemas e da solução

- **Objetivos:** gerar relatórios descrevendo os problemas detectados e indicar a investigação, a análise e a resolução do problema.
- **Artefato de entrada:** plano para resolução de problemas
- **Artefatos de saída:**
 1. Relatório de resolução de problemas
 - 1.1. Descrição do problema detectado
 - 1.2. Categoria e prioridade do problema
 - 1.3. Investigação do problema
 - 1.4. Análise do problema
 - 1.5. Resolução do problema

B.2.8 Processo de Revisão Sistemática

O processo de revisão sistemática é composto pelas atividades de implantação do processo e realização da revisão sistemática.

Implantação do Processo

Identificador: IM

Proposta: desenvolver uma estratégia para a realização de revisão sistemática.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para revisão sistemática seja definida e implantada.

Tarefas:

IM.1 Desenvolver política para revisão sistemática

- **Objetivo:** determinar as atividades de revisão sistemática que serão cumpridas pelas equipes

- Artefato de entrada: documentos indicando exigências em relação à revisão sistemática
- Artefatos de saída:
 1. Plano para revisão sistemática:
 - 1.1. *Guidelines*, procedimentos, metodologias ou critérios que serão usados para revisão sistemática

Realização da revisão sistemática

Identificador: RE

Proposta: auxiliar a avaliação e interpretação de resultados de pesquisas disponíveis relevantes para uma questão de pesquisa, tópico ou fenômeno de interesse.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma avaliação rigorosa e confiável sobre um tópico de pesquisa seja realizada e esteja de acordo com o plano para revisão sistemática.

Tarefas: as tarefas relacionadas à atividade de realização de revisão sistemática foram identificadas a partir do procedimento para execução de revisões sistemáticas apresentadas por Kitchenham (2004). Esse trabalho foi particularmente escolhido por se tratar de uma compilação de outros trabalhos já existentes que indicavam *guidelines* para a realização de revisões sistemáticas para a área de medicina.

RE.1 Planejar a revisão sistemática

- Objetivos: identificar as necessidades de uma revisão sistemática e desenvolver um protocolo para realizá-la.
- Artefatos de entrada: tema para revisão sistemática
- Artefatos de saída:
 1. Planejamento da revisão sistemática
 - 1.1. Objetivos da revisão
 - 1.2. Fontes primárias para identificação de estudos primários
 - 1.3. Critérios de inclusão e exclusão dos estudos
 - 1.4. Critérios utilizados para avaliar a qualidade de estudos primários
 - 1.5. Mecanismo utilizado para extração de dados de estudos primários e para síntese dos dados
 - 1.6. Diferenças entre os estudos investigados
 2. Protocolo para revisão sistemática
 - 2.1. *Background* (razões para realizar a revisão)
 - 2.2. Questão de pesquisa que deve ser respondida
 - 2.3. Estratégia que deve ser utilizada para procurar por documentos

- 2.4. Critérios para incluir e excluir estudos da revisão sistemática
- 2.5. *Checklist* de avaliação de qualidade dos estudos
- 2.6. Estratégia de extração dos dados
- 2.7. Síntese dos dados extraídos

RE.2 Executar a revisão sistemática

- Objetivo: identificar e selecionar estudos, extrair e sintetizar os dados
- Artefatos de entrada: planejamento da revisão sistemática, protocolo para revisão sistemática, documentos científicos
- Artefatos de saída:
 1. Relatório de execução da revisão sistemática
 - 1.1. Resultados da revisão sistemática
 - 1.2. População investigada (periódicos, anais, literatura em geral)
 - 1.3. Resultados da discussão dos estudos com um especialista
 - 1.4. Avaliação da qualidade dos estudos, considerando se a qualidade de cada estudo justifica as diferenças nos resultados, atribuindo pesos a estudos de acordo com suas importâncias
 - 1.5. Informações sobre a revisão: revisor, data, título, autores, detalhes da publicação, notas pessoais do revisor
 - 1.6. Resumo dos resultados dos estudos

RE.3 Registrar os resultados da revisão sistemática

- Objetivos: comunicar os resultados da revisão sistemática
- Artefatos de entrada: planejamento da revisão sistemática, protocolo para revisão sistemática, resultados da revisão sistemática
- Artefatos de saída:
 1. Relatório final da revisão sistemática
 - 1.1. Título
 - 1.2. Autores
 - 1.3. Contexto (a importância da questão de pesquisa tratada na revisão)
 - 1.4. Objetivos (questões tratadas na revisão sistemática)
 - 1.5. *Background* (justificativa da necessidade de revisão)
 - 1.6. Questões de revisão
 - 1.7. Métodos usados na revisão (fontes de dados e estratégia de pesquisa, critérios de seleção de estudos, avaliação de qualidade, extração de dados, síntese de dados)
 - 1.8. Estudos incluídos e excluídos

- 1.9. Resultados
- 1.10. Discussão sobre os principais resultados
- 1.11. Conclusões
- 1.12. Referências e Apêndices

B.2.9 Processo de Preparação de Documentos Científicos

O processo de preparação de documentos científicos é composto pelas atividades de redação de documentos científicos e formatação de documentos científicos.

Redação de documentos científicos

Identificador: RD

Proposta: indicar diretrizes e *guidelines* para auxiliar a redação de documentos científicos, com foco na estrutura. Como os objetivos e a estrutura de documentos científicos (planos de pesquisa, artigos, teses, dissertações, resenhas, etc) diferem muito entre si, não se pode falar em um roteiro rígido para redação desses documentos. No entanto, é possível oferecer um modelo relativamente flexível, que considere os elementos essenciais e que permita incluir itens inerentes ao tipo de documento a ser produzido (Gil, 2002).

Resultados: como resultado da implementação bem sucedida deste processo espera-se que um roteiro que auxilie a redação de documentos científicos seja definido e seja utilizado.

Tarefa: a tarefa relacionada à atividade de redação de documentos científicos foi identificada a partir dos trabalhos desenvolvidos por Bastos et al. (2004), Gonçalves (2004), Walker (1993) e Laville e Dionne (1999).

RD.1 Definir os elementos estruturais do documento

- **Objetivo:** identificar a estrutura que deverá ser utilizada para descrever o trabalho científico. A estrutura deve ser escolhida de acordo com o tipo de documento que deverá ser elaborado.
- **Artefatos de entrada:** texto referente ao conteúdo do documento, normas e padrões indicando elementos estruturais de documentos científicos.
- **Artefatos de saída:**
 1. Estrutura do documento científico:
 - 1.1 Elementos pré-textuais (capa, folha de rosto, folha de aprovação, dedicatória, agradecimentos, epígrafe, resumo, lista de figuras, lista de tabelas, siglas, símbolos e abreviaturas, sumário)
 - 1.2. Elementos textuais (introdução, revisão da literatura, material(is) e

método(s), resultados, discussão e conclusões)

1.3. Elementos pós-textuais (referências, glossário, apêndice(s), anexo(s), índice(s)).

- Informações adicionais, Tabela B.15

Tabela B.15: Informações adicionais relacionadas à tarefa RD.1

Sugestões:
Em particular, em relação à redação de um projeto de pesquisa, Gil (2002) sugere que a estrutura do documento seja composta por introdução (objetivos do trabalho, justificativa, definição e delimitação do problema, revisão da literatura, hipótese(s)); metodologia (tipo de pesquisa, população, amostra, coleta de dados, técnicas para análise de dados); cronograma de execução; suprimentos e equipamentos necessários e custo do projeto.
Alguns trabalhos têm sido desenvolvidos com o objetivo de identificar as estruturas básicas que devem ser consideradas durante a elaboração do resumo de um documento científico (Feltrim, 2004; Teufel et al., 1999). De forma geral, os autores sugerem que sejam apresentadas a contextualização, o problema encontrado na literatura, o propósito, a metodologia, os resultados e as conclusões do trabalho.
Walker (1993) indica alguns <i>guidelines</i> para auxiliar a redação de documentos científicos: devem ser elaborados por pesquisadores novatos e experientes, de forma colaborativa; devem ser disponibilizados modelos (exemplos) de documentos e deve ser criado um “centro de suporte à redação de documentos”. Exemplos desses centros são o <i>Writing Center at Rensselaer Polytechnic Institute</i> e o <i>Writing Assistance Program at North Carolina State</i> .

Formatação de documentos científicos

Identificador: FD

Proposta: indicar diretrizes e *guidelines* para auxiliar a formatação de documentos científicos, com foco na apresentação.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que um roteiro que auxilie a apresentação de documentos científicos seja definido e utilizado.

Tarefa: a tarefa relacionada à atividade de formatação de documentos científicos foi identificada a partir dos trabalhos desenvolvidos por Bastos et al. (2004), Gonçalves (2004) e Laville e Dionne (1999).

FD.1 Apresentar documentos científicos de acordo com regras de formatação

- Objetivo: formatar o documento científico de acordo com regras definidas por normas reconhecidas ou por instituições envolvidas com o desenvolvimento de pesquisas.
- Artefatos de entrada: documento científico (de acordo com o documento de estrutura), normas de apresentação de documentos científicos
- Artefatos de saída:
 1. Documento científico que segue o padrão de formatação escolhido. Foram propostas normas para auxiliar a formatação e apresentação de documentos científicos. Alguns exemplos são a NBR 14724 (Associação Brasileira de Normas Técnicas, 2002a), para apresentação de trabalhos acadêmicos e a NBR 6023 para indicação de referências (Associação Brasileira de Normas Técnicas, 2002b).

B.2.10 Processo de Elaboração de Módulos Educacionais

O processo de elaboração de módulos educacionais é composto pela atividade de elaboração de modelos para representação de conteúdo educacional.

Elaboração de modelos para representação de conteúdo educacional

Identificador: EM

Proposta: indicar diretrizes para o desenvolvimento de módulos educacionais, que poderão ser utilizados na apresentação de trabalhos em conferências, para a realização de treinamento, dentre outros.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que sejam criados modelos para representação do conhecimento e módulos educacionais que auxiliem a transmissão do conhecimento.

Tarefas: as tarefas relacionadas à atividade de elaboração de modelos para representação de conteúdo educacional foram identificadas a partir do trabalho de Barbosa (2004).

EM.1 Determinar os aspectos específicos do domínio de conhecimento

- Objetivo: gerar um modelo conceitual descrevendo, em alto nível, o domínio que se deseja ensinar considerando-se, principalmente, aspectos relativos à estruturação do conhecimento.
- Artefatos de entrada: conceitos e tipos de estruturas que serão utilizadas na organização dos conceitos
- Artefatos de saída:
 1. Modelo conceitual
- Informações adicionais, Tabela B.16

Tabela B.16: Informações adicionais relacionadas à tarefa EM.1

Sugestão:
Barbosa (2004) sugeriu as seguintes atividades para o processo de construção de modelos conceituais: (a) identificar os conceitos relevantes para a compreensão do domínio; (b) identificar relacionamentos de classificação entre os conceitos considerando-se verbos e expressões como “é classificado em” e “é um tipo de”; (c) identificar relacionamentos de composição entre os conceitos considerando-se verbos e expressões como “é formado por”, “é composto de” e “pertence a” e (d) identificar relacionamentos específicos entre os conceitos selecionados. Verbos e expressões tais como “implica em”, “é consequência de”, “utiliza”, “assume” podem ser utilizados.
Comentário:
Conforme indicado por Turine (1998), alguns tipos de estruturas podem ser utilizadas na organização da informação: hierárquica, linear e rede. Como observado por Barbosa (2004), a estruturação hierárquica é a mais enfatizada no contexto educacional. Abordagens para o desenvolvimento de aplicações educacionais que utilizam a estrutura hierárquica foram propostos por Pansanato (1999) e por Pimentel (1997).

EM.2 Refinar o modelo conceitual e definir elementos instrucionais

- Objetivo: gerar um modelo instrucional, identificando informações adicionais que devam ser associadas ao domínio de conhecimento.
- Artefato de entrada: modelo conceitual
- Artefato de saída:
 1. Modelo instrucional
- Informações adicionais, Tabela B.17

EM.3 Estabelecer ordem pedagógica para apresentação das informações

- Objetivo: gerar um ou mais modelos didáticos, estabelecendo formas distintas de utilização e disponibilização de um conteúdo educacional.
- Artefatos de entrada: modelo conceitual, modelo instrucional
- Artefatos de saída:
 1. Modelos didáticos
- Informações adicionais, Tabela B.18

Tabela B.17: Informações adicionais relacionadas à tarefa EM.2

Sugestão:
Barbosa (2004) sugeriu as seguintes atividades para o processo de refinamento do modelo conceitual e construção de modelos instrucionais: (a) para cada conceito representado no modelo conceitual, identificar as informações adicionais (nomes, datas, eventos, conjunto de passos, etapas, etc) que possam ser associadas ao mesmo; (b) associar itens de informação, preservando e estabelecendo a estruturação hierárquica entre eles e (c) definir elementos instrucionais que possam ser associados a cada conceito representado no modelo conceitual: elementos explanatórios, ou seja, informações complementares utilizadas na explicação de um dado item de informação como exemplos, dicas, sugestões de estudo e referências; elementos exploratórios, ou seja, aspectos que permitam explorar o domínio de conhecimento de maneira prática (exercícios guiados, simulações) e elementos de avaliação, ou seja, aspectos que permitam avaliar o aprendizado ocorrido, envolvendo questões na forma objetiva e/ou subjetiva.

Tabela B.18: Informações adicionais relacionadas à tarefa EM.3

Sugestão:
Conforme apresentado por Barbosa (2004), os modelos didáticos são responsáveis pelo estabelecimento de relações de precedência (pré-requisitos) e relacionamentos didáticos, definindo sequências de apresentação entre os objetos caracterizados no modelo instrucional. São considerados relacionamentos didáticos as relações tais como “complementa”, “exemplifica”, “ilustra”, “motiva”, “exercita” e “avalia”. Estruturas de acesso, tais como índices e roteiros também podem ser utilizados. As atividades sugeridas para construção dos modelos didáticos são as seguintes: definir sequências de apresentação entre os objetos modelados; definir relações de precedência entre itens de informação e definir relacionamentos didáticos entre itens de informação e elementos instrucionais.

B.2.11 Processo de *Postmortem*

O processo de *postmortem* é composto pelas atividades de preparação e realização de análise *postmortem* e finalização da análise *postmortem*.

Preparação e realização de análise *postmortem*

Identificador: PE

Proposta: planejar e realizar análise *postmortem*.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a análise *postmortem* seja planejada de forma a ser replicável e que a execução da análise *postmortem* forneça resultados confiáveis.

Tarefas: as tarefas relacionadas à atividade de preparação e realização de análise *postmortem* foram identificadas a partir dos trabalhos de Birk et al. (2002), de Collier et al. (1996) e de Humphrey (1999).

PE.1 Planejamento da análise *postmortem*

- Objetivo: determinar o objetivo, os participantes, as atividades e os recursos necessários para a realização da análise *postmortem*. O tipo de análise *postmortem* também deve ser definido. Conforme observado por Birk et al. (2002), há dois tipos distintos de análise *postmortem*: um deles é aquele considerado mais genérico, que tem como objetivo coletar todo tipo de experiência do grupo e o outro possui o foco mais definido, pois tenta compreender e melhorar uma atividade específica do projeto.
- Artefatos de entrada: objetivo geral da análise *postmortem*, informações relativas a “o que” se deseja obter com a análise *postmortem*
- Artefato de saída:
 1. Plano de análise *postmortem*
 - 1.1. Objetivos
 - 1.2. Participantes, local, data
 - 1.3. Atividades da análise *postmortem*
 - 1.4. Recursos necessários
 - 1.5. Técnicas para análise dos dados
 - 1.6. Mecanismos de divulgação dos resultados
 - 1.7. Tipo de análise *postmortem*
- Informações adicionais, Tabela B.19

PE.2 Coleta de dados

- Objetivos: capturar e registrar experiências dos membros das equipes durante o desenvolvimento do projeto.
- Artefatos de entrada: plano de análise *postmortem*
- Artefatos de saída: são dependentes da abordagem planejada para coleta de dados. Assim, podem ser obtidos como artefatos de saída, resultados de questionários respondidos, registros de fatores positivos, negativos e lições aprendidas mencionados por participantes do projeto em reuniões.

Tabela B.19: Informações adicionais relacionadas à tarefa PE.1

Sugestões:
Para a fase de planejamento da análise <i>postmortem</i> , Collier et al. (1996) sugerem o planejamento e a submissão de questionários quando for necessário coletar informações específicas do projeto sem comprometer a confidencialidade dos participantes. Diferentemente, Desouza et al. (2005) apresentam experiências em que o planejamento de análise de <i>postmortem</i> consistiu na identificação de perguntas que poderiam ser feitas aos participantes em uma reunião, por exemplo, “Quais foram as quatro principais experiências positivas e as quatro principais experiências negativas que você teve durante o desenvolvimento do projeto?”.

Finalização da análise *postmortem*

Identificador: FA

Proposta: analisar os dados e publicar os resultados da análise *postmortem*.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que sejam utilizadas as técnicas para análise dos dados pré-determinadas e que os resultados sejam divulgados para a comunidade de interesse.

Tarefas: as tarefas relacionadas à atividade de finalização da análise *postmortem* foram identificadas a partir dos trabalhos de Birk et al. (2002), de Collier et al. (1996) e de Humphrey (1999).

FA.1 Análise de dados

- Objetivo: sintetizar os dados coletados de forma a auxiliar na elaboração das conclusões.
- Artefatos de entrada: plano de análise *postmortem*, dados coletados
- Artefatos de saída:
 1. Diagramas apresentando os resultados: diferentes diagramas relacionando os principais tópicos mencionados pelos participantes na fase de coleta de dados e os fatores positivos e negativos em relação àquele tópico podem ser gerados. Por exemplo, diagramas Ishikawa (conhecidos como diagrama espinha de peixe) podem ser gerados em um processo colaborativo para auxiliar a encontrar as causas de experiências positivas e negativas e analisar fatores favoráveis e críticos em relação ao projeto.
- Informações adicionais, Tabela B.20

FA.2 Registro e publicação dos resultados

Tabela B.20: Informações adicionais relacionadas à tarefa FA.1

Sugestão:
Conforme sugerido por Collier et al. (1996), caso a submissão de questionários tenha sido utilizada na fase de coleta de dados, pode-se realizar o “dia da história do projeto”, em que alguns participantes do projeto se reúnem com o objetivo de revisar os principais eventos do projeto.

- Objetivos: divulgar os resultados da análise *postmortem* à comunidade de interesse e documentá-los em um relatório de experiências do projeto
- Artefatos de entrada: plano de análise *postmortem*, dados coletados, diagramas gerados
- Artefatos de saída:
 1. Relatório da análise *postmortem*
 - 1.1. Descrição geral do projeto incluindo produtos gerados, métodos utilizados e tempo e esforço necessários
 - 1.2. Principais problemas do projeto mencionados pelos participantes (descrições e diagramas de Ishikawa devem apontar as causas)
 - 1.3. Principais elementos de sucesso dos projetos (de forma semelhante, descrições e diagramas Ishikawa devem ser apresentados)
 - 1.4. Apêndice contendo *transcripts* da análise de *postmortem* (para que os leitores possam compreender como o grupo discutiu os problemas e as experiências de sucesso do projeto)
 - 1.5. Ligações com projetos futuros, que indiquem compromissos dos coordenadores do projeto em mudar e melhorar os processos

B.2.12 Processo de Transferência Tecnológica

O processo de transferência tecnológica é composto pelas atividades de preparação para transferência tecnológica, realização de estudos de caso e implantação do protótipo em setor industrial.

Preparação para transferência tecnológica

Identificador: PT

Proposta: indicar práticas que auxiliam a aplicação de resultados de pesquisas científicas em contexto industrial.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que seja determinado se a transferência tecnológica de resultados de um projeto de pesquisa para o setor industrial é viável.

Tarefas: as tarefas relacionadas à atividade de preparação para transferência tecnológica foram identificadas a partir dos trabalhos de Shull et al. (2001) e de Rombach (2000).

PT.1 Analisar a viabilidade da transferência tecnológica

- **Objetivos:** analisar se o protótipo que se pretende aplicar em contexto industrial cumpre os objetivos relacionados ao seu desenvolvimento e se os esforços necessários para cumprir atividades de transferência tecnológica são compensatórios. Outro objetivo é realizar estudos experimentais para avaliar a efetividade do protótipo (embora não seja estritamente necessário que todas as hipóteses ou questionamentos sejam completamente esclarecidos. Podem ser realizadas, por exemplo, entrevistas com pessoas que tenham utilizado ou avaliado o protótipo considerando-se questões relevantes para as pessoas interessadas).
- **Artefatos de entrada:** protótipo, resultados de entrevistas, resultados de estudos experimentais
- **Artefatos de saída:**
 1. Resultados da análise do protótipo: dados de interesse devem ser coletados e registrados, tais como utilidade prática, benefícios e limitações do protótipo e melhoria na produtividade.
- **Informações adicionais,** Tabela B.21

Tabela B.21: Informações adicionais relacionadas à tarefa PT.1

Comentário:
Shull et al. (2001) propuseram uma metodologia para transferir processos de software que tenham sido definidos e avaliados no setor acadêmico para o setor industrial. O processo proposto pelos autores foi considerado no contexto deste trabalho, no entanto, foi adaptado para o contexto de transferência tecnológica de software. Conforme apresentado por Rost (2005), há um crescente interesse da indústria para que ocorra a transferência de conhecimento, em termos de boas práticas de Engenharia de Software, do setor acadêmico para o industrial. As atividades relacionadas foram tratadas sob a perspectiva de treinamento, Seção B.3.1.

PT.2 Realizar estudo observacional

- **Objetivo:** coletar dados que indiquem como o protótipo é executado. Técnicas de observação podem ser usadas para auxiliar membros da universidade e da indústria a compreender os processos (ou as atividades) que podem ser

executados com auxílio do protótipo. Além disso, podem ser identificados requisitos ou características adicionais que devam ser consideradas caso seja constatada a necessidade de manutenção envolvendo o protótipo. O observador deve, portanto, capturar informações sobre as circunstâncias nas quais as pessoas enfrentam problemas ao lidar com o protótipo.

- Artefatos de entrada: técnicas para realização de estudo observacional, protótipo a ser avaliado
- Artefatos de saída:
 1. Relatório de estudo observacional
 - 1.1. Técnica(s) utilizada(s)
 - 1.2. Dados coletados
 - 1.3. Resultados obtidos
 - 1.4. Conclusões sobre o estudo observacional

Realização de estudos de caso e implantação do protótipo em ambiente industrial

Identificador: RS

Proposta: realizar estudos de caso em ambiente acadêmico e industrial tendo em vista a transferência tecnológica.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que o protótipo seja avaliado, melhorado e implantado em ambiente industrial.

Tarefas: as tarefas relacionadas à atividade de realização de estudos de caso e implantação do protótipo em ambiente industrial foram identificadas a partir dos trabalhos de Shull et al. (2001) e de Rombach (2000).

RS.1 Realizar estudos de caso em ambiente acadêmico

- Objetivo: coletar dados que permitam testar, avaliar ou melhorar o protótipo. A principal motivação para a realização desta atividade é a possibilidade de realizar estudos envolvendo o protótipo em um ambiente em que os custos e os riscos são relativamente baixos.
- Artefatos de entrada: técnicas para realização de estudos de caso, protótipo a ser avaliado
- Artefatos de saída:
 1. Registros dos resultados de estudos de caso em ambiente acadêmico: a estrutura apresentada na norma ISO/IEC 14598-5 (ISO/IEC 14598, 1996) pode ser utilizada para a preparação e a execução de cada estudo de caso realizado. Sugere-se que o documento apresentando os resultados do estudo de

caso apresente as seguintes informações:

- 1.1. Análise de requisitos do estudo de caso (descrição dos objetivos do estudo de caso e indicação dos requisitos avaliados)
- 1.2. Especificação do estudo de caso (definir do escopo do estudo de caso, das medições realizadas, das restrições, dos métodos a serem utilizados e das responsabilidades de todos os envolvidos)
- 1.3. Planejamento do estudo de caso (procedimentos a serem adotados para realização das medições definidas na especificação, identificação de recursos necessários, distribuição desses recursos nas várias ações a serem executadas bem como os prazos, a equipe envolvida, os riscos associados e todas as atividades envolvidas)
- 1.4. Execução do estudo de caso (descrição da execução das atividades planejadas com o objetivo de medir e verificar o artefato de acordo com os requisitos, com a especificação e com o projeto do estudo de caso)
- 1.5. Conclusão do estudo de caso (elaboração, revisão e disponibilização do relatório)

RS.2 Realizar estudos de caso em ambiente industrial

- Objetivo: realizar estudos com um pequeno grupo de pessoas do setor industrial com o objetivo de avaliar o protótipo, ou seja, investigar os benefícios e os problemas da interação com o protótipo no ambiente real em que se pretende utilizá-lo. A avaliação envolve, além do código-fonte, a documentação disponível.
- Artefatos de entrada: técnicas para realização de estudos de caso, protótipo a ser avaliado
- Artefatos de saída:
 1. Registros de resultados sobre estudos de caso no setor industrial: assim como apresentado na descrição da tarefa [RS.1], a norma ISO/IEC 14598-5 pode ser adotada na preparação e execução de estudos de caso no setor industrial.
- Informações adicionais, Tabela B.22

RS.3 Modificar o protótipo e implantar na indústria

- Objetivo: preparar o protótipo para ser utilizado na indústria, realizando as alterações necessárias e disponibilizando os artefatos produzidos.
- Artefatos de entrada: protótipo, registros de resultados sobre estudos de caso
- Artefatos de saída: para a modificação e teste do protótipo deve-se considerar as atividades de processo de desenvolvimento de software cumpridas pela empresa ou pela universidade, considerando-se os interesses, as responsabilidades

Tabela B.22: Informações adicionais relacionadas à tarefa RS.2

Sugestão:
Shull et al. (2001) sugeriram que membros dos setores acadêmico e industrial discutam e avaliem as possibilidades de uso do protótipo no setor industrial. Considerando-se que o pesquisador é o especialista em relação ao protótipo e o membro da indústria é quem de fato conhece as necessidades do setor em que o protótipo será implantado, alguns problemas potenciais podem ser descobertos antes que o protótipo seja adotado pela organização ou por seus próprios clientes.
Comentário:
De acordo com Rombach (1999), há dois objetivos principais em relação ao uso da experimentação em transferência tecnológica: ajuda a persuadir os membros da organização antes de introduzir uma nova tecnologia e, durante o uso da nova tecnologia, a experimentação ajuda na realização das alterações necessárias na tecnologia, promovendo a otimização de seus efeitos. Maldonado et al. (2006) indicaram que pacotes de laboratório são instrumentos que viabilizam a replicação de estudos experimentais e contribuem para a realização de transferência tecnológica. Segundo os autores, os pacotes de laboratório descrevem o experimento em termos específicos, fornecem materiais para replicação, indicam oportunidades para variações e apresentam um contexto que contempla a combinação de resultados considerando diferentes tratamentos.
Experiência:
Resultados de uma experiência relacionada à transferência de tecnologia no contexto de técnicas de leitura de especificação de requisitos de software foram apresentadas por Hohn (2003).

e os acordos definidos para o cumprimento das atividades. Como resultado, deve-se gerar um pacote contendo os artefatos de interesse para a indústria, que pode incluir, por exemplo, o código fonte e a documentação do protótipo, registro de decisões de projeto, lições aprendidas e material para treinamento do protótipo.

B.3 Processos organizacionais

De acordo com a norma ISO/IEC 12207, os processos organizacionais são empregados para estabelecer e implementar uma estrutura constituída pelos processos de ciclo de vida e pelo pessoal envolvido no desenvolvimento do software.

Para o contexto do desenvolvimento de projetos de pesquisa, foram incorporados processos para (1) treinamento; (2) gerência; (3) estabelecimento da infra-estrutura necessária; (4) melhoria do processo cumprido para desenvolvimento de projetos de pesquisa; (5) planejamento; (6) divulgação dos resultados obtidos e das experiências;

(7) comunicação entre os participantes do projeto; (8) coordenação e (9) parceria universidade-empresa.

B.3.1 Processo de Treinamento

O processo de treinamento é composto pelas atividades de implantação do processo e treinamento

Implantação do processo

Identificador: IL

Proposta: preparar pesquisadores e estudantes, em relação às habilidades e aos conhecimentos necessários, para o desenvolvimento de projetos de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para o treinamento seja definida e implantada.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

IL.1 Desenvolver política para treinamento

- Objetivo: determinar as atividades que serão cumpridas e os recursos necessários para a realização do treinamento.
- Artefatos de entrada: documentos indicando exigências em relação ao treinamento, tema do treinamento
- Artefato de saída:
 1. Plano para treinamento
 - 1.1. Atividades de treinamento
 - 1.2. Recursos necessários (equipamentos, software, infra-estrutura, etc)

Treinamento

Identificador: TR

Proposta: fornecer treinamento aos participantes do desenvolvimento do projeto de pesquisa de forma a assegurar que eles tenham as habilidades requeridas para executar as atividades propostas.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as equipes que participam do desenvolvimento do projeto de pesquisa sejam treinadas de acordo com suas necessidades.

Tarefas: as tarefas relacionadas à atividade de treinamento foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

TR.1 Treinar equipes

- **Objetivos:** elaborar material para a realização do treinamento, cumprir as atividades de treinamento planejadas e manter registros de treinamento. A elaboração de material envolve a preparação de módulos educacionais, conforme apresentado no processo de elaboração de módulos educacionais (Seção B.2.10).
- **Artefato de entrada:** plano para treinamento
- **Artefatos de saída:**
 1. Material de treinamento
 2. Registros de treinamento
 - 2.1. Nome do(s) instrutor(es)
 - 2.2. Nome dos participantes
 - 2.3. Título, data, tempo de treinamento
 - 2.4. Resumo do tema abordado
 - 2.5. Solicitação dos participantes de treinamentos adicionais
- **Informações adicionais,** Tabela B.23

Tabela B.23: Informações adicionais relacionadas à tarefa TR.1

Sugestão:
Bleek et al. (2005) indicaram atividades que estão sendo cumpridas de forma bem sucedida para treinamento em ambiente industrial e que, segundo ele, poderiam ser cumpridas também em universidades. Em resumo, os autores sugerem que (1) sejam apresentadas aulas teóricas intercaladas com aulas práticas em laboratório; (2) sejam realizados pequenos ciclos de produção e <i>feedback</i> em que artefatos de interesse são produzidos pelos participantes, avaliados e comentados pelos responsáveis pelo treinamento; (3) seja desenvolvido um mini-projeto de forma que os participantes possam perceber o uso da tecnologia ou do processo que está em treinamento no contexto do desenvolvimento de um projeto e (4) sejam oferecidas contribuições em um projeto real e não crítico, de forma que os participantes possam experimentar problemas típicos do desenvolvimento de projetos de software em diferentes níveis: sociais, técnicos e organizacionais.
Comentário:
Conforme sugerido na Norma ISO/IEC 15504-5, ao final do treinamento, os participantes devem avaliar a qualidade do treinamento oferecido, tendo em vista a melhoria do processo utilizado. Uma possibilidade, neste caso, é a realização de uma análise <i>postmortem</i> (Seção B.20).

B.3.2 Processo de Gerenciamento

O processo de gerenciamento é composto pelas atividades de gerenciamento de escopo, gerenciamento de recursos humanos, gerenciamento da capacidade das equipes, gerenciamento de atividades, gerenciamento de cronograma, gerenciamento de custos, gerenciamento de riscos e gerenciamento de requisitos.

Gerenciamento de escopo

Identificador: GE

Proposta: gerenciar as diferentes equipes de desenvolvimento do projeto de pesquisa em relação ao escopo do projeto.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que o escopo do projeto seja gerenciado.

Tarefas: a tarefa relacionada à atividade de gerenciamento de escopo foi identificada a partir da norma ISO/IEC 15504-5.

GE.1 Gerenciamento de escopo

- **Objetivos:** gerenciar as mudanças que ocorrem no escopo do projeto de pesquisa e efetuar mudanças no documento de definição do escopo quando necessário.
- **Artefato de entrada:** documento de definição do escopo, solicitações de mudanças
- **Artefato de saída:**
 1. Documento de definição do escopo atualizado, refletindo as alterações. É fundamental identificar as causas das mudanças no planejamento do escopo; haver um acordo por parte de todas as equipes em relação às mudanças solicitadas e monitorar as mudanças.

Gerenciamento de recursos humanos

Identificador: GR

Proposta: auxiliar a organização e o gerenciamento das equipes de trabalho em um ambiente em que há alta rotatividade de pessoas.

Resultados: como resultado da implementação bem sucedida deste processo espera-se que habilidades e competências necessárias para o desenvolvimento do projeto sejam identificadas e que o gerente do projeto e os coordenadores locais sejam capazes de coordenar a alocação de tarefas de acordo com a disponibilidade dos recursos humanos.

Tarefas: as tarefas relacionadas à atividade de gerenciamento de recursos humanos foram identificadas a partir do PMBOK, da norma ISO/IEC 15504-5 e dos trabalhos de Maidantchik (1999), Acuna e Juristo (2004) e Chiavenato (2002).

GR.1 Preparar para o recrutamento

- **Objetivos:** estabelecer as necessidades do projeto que levam ao recrutamento de novos membros e definir as técnicas de recrutamento e os critérios de seleção.
- **Artefatos de entrada:** artefatos gerados nas fases de análise de requisitos do sistema, projeto arquitetural do sistema e elaboração de plano de pesquisa, por apresentarem uma visão do sistema (mais geral) e do projeto (mais específico). Os artefatos são relevantes no contexto desta tarefa porque auxiliam o entendimento de partes do projeto para as quais há a necessidade de recrutamento.
- **Artefatos de saída:**
 1. Necessidade de recrutamento
 - 1.1. Número de pessoas que serão recrutadas
 - 1.2. Perfil técnico desejado
 - 1.3. Perfil não técnico desejado
 - 1.4. Justificativa da necessidade de recrutamento
 2. Técnicas de recrutamento e critérios de seleção
 - 2.1. Técnica(s) utilizada(s) no recrutamento
 - 2.2. Meio utilizado para recebimento de propostas dos interessados, por exemplo, por email ou entrega de documentos impressos pessoalmente
 - 2.3. Critérios de seleção dos candidatos
- **Informações adicionais,** Tabela B.24

GR.2 Preparar e tornar pública a chamada à participação no desenvolvimento de projetos de pesquisa

- **Objetivos:** elaborar um documento ou um texto informal apresentando a possibilidade de participação de novo(s) membro(s) no projeto de pesquisa, a estratégia de seleção e o perfil desejado dos candidatos. Além disso, deve-se publicar a chamada à participação.
- **Artefatos de entrada:** necessidade de recrutamento, técnicas de recrutamento e critérios de seleção.
- **Artefato de saída:**
 1. Chamada à participação no desenvolvimento de projetos de pesquisa

Tabela B.24: Informações adicionais relacionadas à tarefa GR.1

Comentários:
As técnicas de recrutamento apresentadas por Chiavenato (2002) podem ser consideradas no contexto de desenvolvimento de pesquisas e se referem, por exemplo, à análise de currículo de candidatos que se apresentaram em outros recrutamentos, apresentação de candidatos por outros estudantes e pesquisadores, divulgação de anúncios e contatos com outros pesquisadores.
De acordo com Chiavenato, as três fases do planejamento do recrutamento são: realizar pesquisa interna (identificar o que a organização ou o grupo precisa), realizar pesquisa externa (identificar o que o mercado ou a comunidade pode oferecer) e escolher técnicas de recrutamento. Em relação à segunda fase, é interessante considerar os comentários de Dannelly e Steidley (2002). Os autores comentam que muitos estudantes que podem oferecer importantes contribuições aos projetos de pesquisa não conhecem os trabalhos que estão sendo desenvolvidos no próprio departamento em que estão associados. Por outro lado, devido à alta rotatividade dos estudantes nos cursos, os coordenadores dos projetos nem sempre conhecem seus perfis e habilidades. Assim, acredita-se que a realização de apresentações breves sobre os projetos de pesquisa em disciplinas dos cursos, na forma de seminários ou palestras, seja um mecanismo importante para a divulgação dos projetos e aumente a possibilidade de que os estudantes se interessem por participar do desenvolvimento de projetos de pesquisa.
Gil (2002) e Goldenberg (2003) indicaram qualidades pessoais que um pesquisador deve ter e que podem ser consideradas para auxiliar a definição do perfil não técnico dos membros que serão recrutados: conhecimento do assunto a ser pesquisado, curiosidade, criatividade, integridade intelectual, atitude autocorretiva, sensibilidade social, imaginação disciplinada, perseverança e paciência, confiança na experiência e capacidade de articular teoria e dados empíricos (analisar os dados coletados em relação a um corpo de conhecimento acumulado por outros estudiosos).

- 1.1. Nome do projeto
- 1.2. Coordenador de projeto e instituição
- 1.3. Descrição do projeto
- 1.4. Indicação de atividade(s) a ser(em) cumprida(s) pelo(s) novo(s) membro(s)
- 1.5. Perfil desejado
- 1.6. Etapas do processo de seleção, incluindo datas e locais.

- Informações adicionais, Tabela B.25

GR.3 Avaliar perfil dos candidatos e realizar seleção

Tabela B.25: Informações adicionais relacionadas à tarefa GR.2

Comentários:
Potenciais candidatos devem avaliar seu perfil e sua disponibilidade para participar do projeto considerando, por exemplo, elementos apresentados na norma ISO/IEC 15504: cumprimento aos requisitos apresentados na chamada à participação e restrições de tempo (ou outras restrições) que possam impedir a participação.

- **Objetivos:** avaliar cada uma das propostas submetidas em relação aos critérios estabelecidos utilizando a técnica de recrutamento definida e selecionar membros que participarão do desenvolvimento do projeto de pesquisa.
- **Artefatos de entrada:** necessidade de recrutamento, técnicas de recrutamento e critérios de seleção e documentos fornecidos pelos candidatos
- **Artefatos de saída:**
 1. Avaliação do perfil dos candidatos
 - 1.1. Nome
 - 1.2. Cumprimento aos critérios estabelecidos
 - 1.3. Fatores adicionais que motivam a seleção do candidato
 2. Classificação dos candidatos
 - 2.1. Candidatos classificados
 - 2.2. Candidatos selecionados

GR.4 Negociar e assinar termo de responsabilidades

- **Objetivos:** negociar e aprovar um termo de responsabilidades (quando pertinente devem assinar um contrato) indicando claramente as expectativas, os compromissos e os deveres do(s) candidato(s) selecionado(s).
- **Artefatos de entrada:** termo de responsabilidades, contrato (sugeridos por agência financiadora ou setor industrial)
- **Artefatos de saída:** diferentes resultados podem ser obtidos com o cumprimento desta tarefa. Quando há uma agência financiando um projeto ou parceria com o setor industrial, é comum que exista um contrato padronizado (Bleek et al., 2005). Em outras situações, o próprio coordenador do projeto pode redigir um contrato. Neste caso, conforme sugerido na norma 15504, pode ser elaborado um contrato conforme apresentado a seguir:
 1. Contrato para trabalho em projeto de pesquisa
 - 1.1. Escopo do contrato
 - 1.2. Prazo de validade do contrato

- 1.3. Riscos e contingências possíveis
- 1.4. Alinhamento do contrato com o valor estratégico do projeto
- 1.5. Proteção à informação proprietária
- 1.6. Produtos a serem entregues pelo estudante/pesquisador selecionado e informações de licenças, *copyright* e patentes relacionadas aos protótipos
- 1.7. Padrões e procedimentos a serem usados
- 1.8. Recebimento de bolsa e valor
- 1.9. Identificação de papéis
- 1.10. Carga horária semanal de trabalho do estudante/pesquisador selecionado

GR.5 Registrar informações gerais sobre membros

- Objetivo: registrar dados dos membros que irão compor as equipes de desenvolvimento do projeto de pesquisa
- Artefato de entrada: não há
- Artefato de saída:
 1. Identificação da equipe
 - 1.1. Identificação
 - 1.2. Universidade/empresa em que está associado(a)
 - 1.3. Email (ou outras informações para contato)
 - 1.4. Áreas de interesse
 - 1.5. Experiências anteriores

Gerenciamento das capacidades das equipes

Identificador: GC

Proposta: garantir que as atividades a serem delegadas às equipes sejam adequadas às suas capacidades

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que informações sobre as habilidades de cada equipe sejam mantidas em um repositório, sejam utilizadas para a atribuição de atividades a cada equipe quando necessário e sejam atualizadas.

Tarefas: as tarefas relacionadas à atividade de gerenciamento das capacidades das equipes foram identificadas a partir do trabalho de Acuna e Juristo (2004) e Paasvaara e Lassenius (2004).

GC.1 Determinar as capacidades de cada equipe

- Objetivo: fazer o mapeamento entre os membros de cada equipe e suas capacidades.

- Artefatos de entrada: identificação dos membros das equipes, capacidades a serem avaliadas.
- Artefatos de saída: deve ser realizada uma avaliação das capacidades de cada uma das pessoas envolvidas no desenvolvimento do projeto de pesquisa por meio de entrevistas, reuniões, *workshops* e *brainstormings*.
 1. Avaliação das capacidades
 - 1.1. Identificação do participante
 - 1.2. Descrição geral das atividades realizadas para determinação das capacidades
 - 1.3. Capacidades identificadas
- Informações adicionais, Tabela B.26

Tabela B.26: Informações adicionais relacionadas à tarefa GC.1

Sugestão:
Acuna e Juristo (2004) indicaram “capacidades de comportamento” importantes para o desenvolvimento de projetos de software: análise de problemas, tomada de decisão, independência, inovação/criatividade, análise crítica, trabalho sob pressão, organização, gerenciamento de riscos, entendimento das condições da organização, disciplina, atendimento a clientes, habilidade de negociação, empatia, sociabilidade, trabalho em grupo/cooperação, avaliação, liderança e planejamento

GC.2 Associar pessoas a papéis e responsabilidades

- Objetivo: determinar os papéis e as responsabilidades que cada uma das pessoas envolvidas com o desenvolvimento do projeto de pesquisa deve assumir.
- Artefato de entrada: avaliação das capacidades, lista de atividades do projeto, modelo de mapeamento entre papéis e capacidades necessárias (um exemplo de modelo foi indicado por Acuna e Juristo (2004))
- Artefato de saída: de forma geral, deve-se observar o quanto determinado membro está qualificado ou tem condições de assumir um papel no contexto do projeto. Como resultado, deve-se alocar as atividades aos membros das equipes, conforme sugerido por Paasvaara e Lassenius (2004) (Tabela B.27).

Gerenciamento de atividades

Identificador: GT

Proposta: gerenciar as diferentes equipes de desenvolvimento do projeto de pesquisa, em relação ao cumprimento das atividades propostas. De acordo com Bernat et al.

Tabela B.27: Alocação de papéis e responsabilidades aos membros

Membro	Papel	Responsabilidades

(2000), é fundamental “quebrar” os objetivos de pesquisa em atividades bem definidas e atribuí-las aos estudantes e pesquisadores, fazendo um mapeamento explícito entre as pessoas envolvidas com o projeto de pesquisa e com as atividades.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as atividades cumpridas pelas equipes sejam gerenciadas.

Tarefa: a tarefa relacionada à atividade de gerenciamento de atividades foi identificada a partir da norma ISO/IEC 15504-5 e do trabalho de Maidantchik (1999).

GT.1 Gerenciamento de atividades das equipes

- **Objetivo:** monitorar o cumprimento de atividades técnicas pelas equipes e efetuar mudanças na associação entre pessoas e atividades quando necessário.
- **Artefatos de entrada:** alocação de papéis e responsabilidades aos membros, informações sobre o cumprimento de atividades e o desempenho dos trabalhos (o coordenador de cada equipe observa diretamente o desempenho dos membros da equipe), solicitações de mudanças.
- **Artefatos de saída:** conforme indicado no documento do PMBOK, o gerente deve monitorar indicadores como progresso em relação às entregas do projeto (pontualidade e qualidade) e soluções adotadas para resolução de problemas. A necessidade de avaliações e revisões depende da extensão e complexidade do projeto, da política organizacional, dos requisitos do contrato e da quantidade e da qualidade da comunicação. As revisões devem ser solicitadas caso necessário. A avaliação dos membros das equipes pode indicar a necessidade de mudanças ou atualizações na associação de pessoas a papéis e a responsabilidades, a descoberta de problemas, o desenvolvimento de planos de treinamento individuais e o estabelecimento de metas específicas para o futuro.
- **Informações adicionais,** Tabela B.28

Gerenciamento de cronograma

Identificador: GO

Proposta: gerenciar as diferentes equipes de desenvolvimento do projeto de pesquisa em relação ao cumprimento do cronograma proposto.

Tabela B.28: Informações adicionais relacionadas à tarefa GT.1

Sugestão:
No documento do PMBOK é sugerido o registro de lições aprendidas no contexto de gerenciamento de recursos humanos, ou seja, devem ser armazenados dados históricos incluindo: organogramas do projeto contendo descrições de cargos e planos de gerenciamento; regras básicas, técnicas de gerenciamento de conflitos e eventos de reconhecimento que foram especialmente úteis; procedimentos para gerenciamento de equipes virtuais, negociação, treinamento e formação da equipe que foram comprovadamente bem-sucedidos; habilidades ou competências especiais de membros da equipe que foram descobertas durante o projeto e problemas encontrados e soluções adotadas.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que o cronograma das equipes seja gerenciado.

Tarefa: a tarefa relacionada à atividade de gerenciamento de cronograma foi identificada a partir da norma ISO/IEC 15504-5 e do trabalho de Maidantchik (1999).

GO.1 Gerenciamento de cronograma das equipes

- Objetivo: avaliar o desenvolvimento do projeto em termos de cumprimento de prazos e efetuar mudanças no cronograma quando necessário.
- Artefato de entrada: cronograma do projeto, solicitações de mudanças
- Artefato de saída:
 1. Cronograma atualizado, refletindo as alterações que ocorreram no projeto em relação a prazos. Uma atualização no cronograma é qualquer modificação nas informações que constam no modelo do cronograma que é usado para gerenciar o projeto (as alterações podem ser necessárias em cronogramas de cada equipe ou no cronograma do projeto). As partes interessadas são notificadas das mudanças conforme são realizadas. As solicitações de mudanças podem ocorrer formal ou informalmente, de acordo com as características de gerenciamento do projeto.
- Informações adicionais, Tabela B.29

Gerenciamento de custos

Identificador: GS

Proposta: gerenciar as diferentes equipes de desenvolvimento do projeto de pesquisa em relação aos custos do projeto.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que os custos para o desenvolvimento do projeto de cada equipe sejam gerenciados.

Tabela B.29: Informações adicionais relacionadas à tarefa GO.1

Comentário:
Conforme observado por Gil (2002), o cronograma corresponde apenas a uma estimativa de tempo para a obtenção dos resultados de pesquisa. Por uma série de fatores imprevistos, os prazos podem deixar de ser observados. Portanto, o registro desses fatores pode ser considerado importante para ajudar na elaboração de cronogramas de novos projetos (lições aprendidas ajudam a melhorar a forma como se faz um cronograma).

Tarefa: a tarefa relacionada à atividade de gerenciamento de custos foi identificada a partir da norma ISO/IEC 15504-5 e do trabalho de Maidantchik (1999).

GS.1 Gerenciamento de custos das equipes

- **Objetivo:** gerenciar os fatores que criam variações nos custos do projeto de pesquisa e efetuar mudanças na planilha de custos do projeto quando necessário.
- **Artefato de entrada:** orçamento do projeto, solicitações de mudanças
- **Artefato de saída:**
 1. Planilha de custos do projeto atualizada, refletindo as alterações que ocorreram em relação aos custos. É fundamental identificar as causas das mudanças no planejamento dos custos; haver um acordo por parte de todas as equipes em relação às mudanças solicitadas, monitorar as mudanças reais quando e conforme ocorrem, garantir que os pedidos de mudança nos custos não ultrapassem o financiamento autorizado periodicamente e o total para o projeto e monitorar o desempenho de custos para detectar e compreender as variações em relação ao planejamento dos custos.
- **Informações adicionais,** Tabela B.30

Tabela B.30: Informações adicionais relacionadas à tarefa GS.1

Comentário:
Conforme apresentado no documento do PMBOK, as lições aprendidas sobre as mudanças no custo do projeto devem ser documentadas. O objetivo é compor um banco de dados históricos tanto para o projeto quanto para a organização executora. A documentação das lições aprendidas inclui as causas das variações, os motivos pelos quais as ações corretivas foram escolhidas e os outros tipos de lições aprendidas de controle de produção de recursos e custos.

Gerenciamento de riscos

Identificador: GI

Proposta: gerenciar riscos que possam comprometer o desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que os riscos identificados para o desenvolvimento do projeto sejam gerenciados.

Tarefas: a tarefa relacionada à atividade de gerenciamento de riscos foi identificada a partir da norma ISO/IEC 15504-5.

GI.1 Gerenciamento de riscos das equipes

- **Objetivos:** monitorar a ocorrência dos riscos identificados, realizar nova análise dos riscos existentes, monitorar as condições de acionamento do plano de resposta aos riscos e revisar a eficácia da execução de respostas a riscos.
- **Artefatos de entrada:** registro de riscos identificados, solicitações de mudanças aprovadas (solicitações informais devem ser documentadas antes que as alterações nos riscos identificados sejam realizadas)
- **Artefato de saída:**
 1. **Identificação e análise de riscos atualizados:** o gerenciamento de riscos exige a constante identificação de novos riscos e a reavaliação de riscos documentados. As reavaliações de riscos do projeto devem ser realizadas regularmente. O gerenciamento de riscos do projeto pode ser um item da pauta das reuniões periódicas de andamento do desenvolvimento do projeto de pesquisa. Os coordenadores locais são responsáveis pelo gerenciamento dos riscos de cada equipe e devem comunicar as alterações ao coordenador do projeto, que é responsável pelo monitoramento global dos riscos do projeto.

Gerenciamento de requisitos

Identificador: GN

Proposta: garantir que os requisitos do software sejam implementados durante o desenvolvimento do projeto de pesquisa e gerenciar as mudanças nos requisitos.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que os requisitos do projeto de pesquisa, alocados às equipes, sejam gerenciados.

Tarefas: a tarefa relacionada à atividade de gerenciamento de requisitos foi identificada a partir do trabalho de Maidantchik (1999).

GN.1 Gerenciamento de requisitos do projeto

- **Objetivos:** analisar solicitações das equipes de projeto em relação a pedidos de mudança de requisitos e identificar possíveis modificações no planejamento,

desenvolvimento, produtos intermediários, atividades, recursos, riscos, custos e outros elementos.

- Artefatos de entrada: requisitos do projeto, solicitações de mudanças, artefatos que serão alterados devido às mudanças nos requisitos (por exemplo, planejamento do projeto, lista de atividades, etc)
- Artefato de saída:
 1. Documento de requisitos (ou outros artefatos descrevendo requisitos) atualizado. Outros artefatos relevantes também devem ser atualizados. Inconsistências identificadas devem ser registradas, bem como as decisões tomadas.

B.3.3 Processo de Infra-estrutura

O processo de infra-estrutura é composto pelas atividades de implantação do processo e implantação de infra-estrutura.

Implantação do processo

Identificador: IT

Proposta: identificar a infra-estrutura necessária para o desenvolvimento do projeto de pesquisa e planejar sua instalação.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a infra-estrutura necessária seja identificada e a instalação seja planejada.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

IT.1 Definir recursos de infra-estrutura e planejar instalação

- Objetivo: identificar equipamentos de hardware e software, ferramentas, padrões (por exemplo, normas e documentos), elementos de segurança, recursos para acesso remoto, *backup* e recuperação, recursos do ambiente de desenvolvimento e outros recursos que sejam importantes para o desenvolvimento, a operação e a manutenção do projeto de pesquisa.
- Artefato de entrada: requisitos de infra-estrutura
- Artefatos de saída:
 1. Relatório de infra-estrutura
 - 1.1. Recurso de infra-estrutura
 - 1.2. Quantidade
 - 1.3. Especificações
 - 1.4. Justificativa

2. Planejamento da instalação
 - 2.1. Recurso
 - 2.2. Local
 - 2.3. Data prevista
 - 2.4. Responsáveis
 - 2.5. Atividades para integração de recursos de infra-estrutura (inclusão de estações de trabalho e servidores em uma rede, a configuração de ferramentas de software, etc)

IT.2 Estabelecer políticas de uso da infra-estrutura

- Objetivo: estabelecer permissões e proibições em relação ao uso da infra-estrutura
- Artefatos de entrada: políticas institucionais (da universidade, da agência financiadora, da indústria parceira) sobre uso de infra-estrutura
- Artefatos de saída:
 1. Políticas de uso
 - 1.1. Permissões de uso
 - 1.2. Direitos de acesso
 - 1.3. Cuidados e preservação
 - 1.4. Prioridades
 - 1.5. Problemas com o funcionamento dos recursos e ações a serem tomadas
 - 1.6. Informações adicionais sobre o problema, ações realizadas e responsável pelas ações
 - 1.7. Prioridade na resolução de problemas relativos aos recursos
 - 1.8. Proibições em relação ao uso dos recursos

Implantação de infra-estrutura

Identificador: IF

Proposta: instalar recursos de infra-estrutura e preparar para manutenção

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que a infra-estrutura seja instalada e as atividades para manutenção dos recursos sejam planejadas.

Tarefa: a tarefa relacionada à atividade de implantação da infra-estrutura foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

IF.1 Instalar e planejar a manutenção dos recursos de infra-estrutura

- **Objetivos:** preparar o ambiente para o desenvolvimento do projeto de pesquisa em relação aos recursos de infra-estrutura e indicar as atividades para manutenção dos recursos.
- **Artefatos de entrada:** planejamento da instalação
- **Artefatos de saída:**
 1. Relatório de instalação de infra-estrutura
 - 1.1. Recursos
 - 1.2. Data de instalação
 - 1.3. Local de instalação
 - 1.4. Informações sobre garantias
 2. Políticas para manutenção dos recursos
 - 2.1. Prioridades
 - 2.2. Medidas para solicitação de manutenção
 - 2.3. Medidas a serem tomadas para realizar manutenção

B.3.4 Processo de Melhoria

O processo de melhoria é composto pelas atividades de determinação do processo, avaliação do processo e melhoria do processo.

Determinação do processo

Identificador: DR

Proposta: identificar os processos que serão cumpridos para o desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que os processos que serão cumpridos sejam descritos, que a descrição seja atualizada sempre que necessário e que a aplicabilidade de cada processo seja indicada. As atividades, as tarefas e os artefatos associados ao processo padrão cumprido e as expectativas em relação ao desempenho também devem ser indicadas.

Tarefas: a tarefa relacionada à atividade de determinação do processo foi identificada a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

DR.1 Descrever e manter a descrição de cada processo cumprido

- **Objetivo:** apresentar as características dos processos cumpridos, atualizando as informações sempre que necessário.
- **Artefatos de entrada:** documentos e relatórios que indiquem como o processo está sendo cumprido

- Artefato de saída:
 1. Descrição do processo
 - 1.1. Atividades do processo
 - 1.2. Utilidade do processo (por que deve ser cumprido)
 - 1.2. Tarefas
 - 1.3. Entradas e saídas para as tarefas
 - 1.4. Critérios de entrada e saída para as tarefas
 - 1.5. Pontos de controle no processo, em que revisões são realizadas e decisões são tomadas
 - 1.6. Interfaces externas com outros processos
 - 1.7. Medições do processo

Avaliação do processo

Identificador: AO

Proposta: determinar se os processos cumpridos contribuem para que sejam alcançados os objetivos do desenvolvimento de pesquisas nas instituições parceiras e indicar a necessidade de melhoria no processo.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que informações relacionadas aos aspectos positivos e negativos dos processos cumpridos sejam conhecidos e contribuam para a melhoria dos mesmos.

Tarefas: as tarefas relacionadas à atividade de avaliação do processo foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

AO.1 Definir plano de avaliação

- Objetivos: elaborar e documentar um plano para avaliação dos processos cumpridos para o desenvolvimento dos projetos de pesquisa.
- Artefatos de entrada: modelos para avaliação de processos de software, alocação de papéis e responsabilidades aos membros, documento de requisitos do sistema
- Artefatos de saída:
 1. Plano de avaliação do processo
 - 1.1. Identificação do patrocinador
 - 1.2. Objetivos da avaliação
 - 1.3. Escopo da avaliação
 - 1.4. Características dos protótipos (domínio de aplicação, complexidade, características gerais)
 - 1.5. Restrições de avaliação (limitações de tempo e recursos, processos a serem

excluídos)

- 1.6. Propriedade dos resultados de avaliação e restrições em relação ao uso
- 1.7. Modelos usados na avaliação
- 1.8. Identidade dos avaliadores
- 1.9. Critérios de competência dos avaliadores
- 1.10. Responsabilidades na realização da avaliação

AO.2 Realizar avaliação

- Objetivo: coletar os dados requeridos para avaliar o processo considerando o escopo do plano de avaliação do processo
- Artefato de entrada: plano de avaliação do processo
- Artefato de saída:
 1. Relatório de avaliação de processo
 - 1.1. Processo avaliado
 - 1.2. Descrição geral dos resultados
 - 1.3. Problemas identificados
 - 1.4. Causas dos problemas identificados
 - 1.5. Evidências objetivas da execução do processo
 - 1.6. Sugestão para correção dos problemas
 - 1.7. Benefícios que podem ser obtidos com a melhoria do processo
 - 1.8. Informações adicionais que sejam úteis para apoiar a melhoria do processo
 - 1.9. Data da avaliação
 - 1.10. Identificação dos avaliadores

Melhoria do processo

Identificador: MP

Proposta: melhorar continuamente o processo utilizado pelas instituições envolvidas no desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que objetivos de melhoria sejam identificados e priorizados e mudanças para o processo sejam definidas e implementadas.

Tarefas: as tarefas relacionadas à atividade de melhoria do processo foram identificadas a partir das normas ISO/IEC 15504 e ISO/IEC 12207.

MP.1 Estabelecer objetivos de melhoria

- Objetivos: analisar como o processo está sendo executado correntemente, focalizando processos para os quais há estímulo para melhoria, resultando na identificação dos objetivos de melhoria.

- Artefatos de entrada: relatório de avaliação de processo
- Artefatos de saída: para a definição dos objetivos de melhoria de processos, podem ser realizadas reuniões, *workshops* e *brainstormings* entre membros das equipes que participam do desenvolvimento.
 1. Objetivos de melhoria de processo
 - 1.1. Objetivos identificados
 - 1.2. Prioridades

MP.2 Melhorar processos

- Objetivos: cumprir as atividades que levem à melhoria dos processos, considerando os objetivos de melhoria estabelecidos.
- Artefatos de entrada: relatório de avaliação de processo, objetivos de melhoria de processo, modelos de melhoria de processos de software
- Artefatos de saída:
 1. Registros de melhoria de processo
 - 1.1. Atividades cumpridas
 - 1.2. Identificação dos responsáveis pelo cumprimento das atividades de melhoria
 - 1.3. Experiências e lições aprendidas

B.3.5 Processo de Planejamento

O processo de planejamento é composto pelas atividades de definição de escopo do projeto, planejamento de atividades, planejamento de prazos, planejamento de custos e planejamento de riscos.

Definição de escopo do projeto

Identificador: DE

Proposta: delimitar a pesquisa a ser realizada. De forma geral, define-se quais elementos especificados nas atividades de análise de requisitos do sistema e projeto arquitetural do sistema serão considerados no projeto de pesquisa. De acordo com Marconi e Lakatos (2002), uma pesquisa pode ser delimitada em relação ao assunto (seleção de um tópico, evitando que a pesquisa se torne muito extensa); à extensão (nem sempre é possível abranger todo o contexto no qual o problema formulado se insere) e a outros fatores (econômicos, ambientais, etc).

Resultados: como resultado da implementação bem sucedida desta atividade espera-se o escopo do projeto de pesquisa seja definido.

Tarefas: as tarefas relacionadas à atividade de definição do escopo do projeto foram identificadas a partir do PMBOK.

DE.1 Identificar escopo inicial

- **Objetivos:** determinar os limites do trabalho no projeto e identificar os resultados que se pretende obter com o desenvolvimento do projeto.
- **Artefatos de entrada:** formulação do problema, documento de requisitos do sistema, documento de análise do sistema, projeto arquitetural do sistema
- **Artefato de saída:**
 1. Documento de definição do escopo
 - 1.1. Objetivo geral da proposta do projeto
 - 1.2. Declaração de escopo
 - 1.3. Justificativa para delimitação do escopo
 - 1.4. Resultados principais (*deliverables*)
 - 1.5. Objetivos do projeto considerando a delimitação do escopo
 - 1.6. Registro das alterações de escopo e justificativa das alterações

DE.2 Criar EAP (Estrutura Analítica do Projeto)

- **Objetivos:** subdividir o trabalho do projeto em partes menores e mais facilmente gerenciáveis, em que cada nível descendente da EAP represente uma definição cada vez mais detalhada do trabalho do projeto.
- **Artefato de entrada:** documento de definição do escopo
- **Artefatos de saída:**
 1. EAP para o projeto de pesquisa
- **Informações adicionais,** Tabela B.31

Planejamento de atividades

Identificador do processo: PI

Proposta: planejar as atividades que serão cumpridas para o desenvolvimento projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as atividades necessárias para o desenvolvimento do projeto de pesquisa sejam identificadas.

Tabela B.31: Informações adicionais relacionadas à tarefa DE.2

Comentário:
Os componentes que compõem a EAP auxiliam na visualização dos produtos que deverão ser entregues com o desenvolvimento do projeto. É comum que detalhes da EAP sejam definidos ou refinados gradualmente, a medida que o projeto se torna mais claro. Em desenvolvimento de projetos de pesquisa nem sempre é possível estabelecer quais serão os artefatos resultantes no início do desenvolvimento do projeto pois, em geral, os requisitos e as definições são estabelecidas de forma iterativa, a medida que se desenvolve o projeto. Esta técnica é conhecida como planejamento em ondas sucessivas.

Tarefas: as tarefas relacionadas ao processo de planejamento de atividades foram identificadas a partir do PMBOK, da norma ISO/IEC 15504-5 e dos trabalhos de Maidantchik (1999) e de Humphrey (1999).

PI.1 Identificação de atividades de projeto

- Objetivo: identificar as atividades que devem ser cumpridas para que o projeto de pesquisa seja desenvolvido.
- Artefatos de entrada: escopo do projeto, EAP (os pacotes de trabalho do projeto identificados durante a elaboração da EAP são decompostos em atividades menores, que são as atividades do projeto de pesquisa), documento de requisitos.
- Artefatos de saída:
 1. Lista de atividades do projeto de pesquisa
 2. Lista de atividades de cada equipe

PI.2 Sequenciamento de atividades

- Objetivo: identificar os relacionamentos lógicos entre as atividades e definir quais atividades podem ser cumpridas em paralelo, por diferentes equipes. É importante identificar também as atividades que serão desenvolvidas de forma colaborativa. As atividades podem ser seqüenciadas logicamente usando relações de precedência adequadas, além de antecipações e atrasos, para dar suporte ao desenvolvimento posterior de um cronograma do projeto de pesquisa.
- Artefatos de entrada: lista de atividades do projeto de pesquisa, lista de atividades de cada equipe

- Artefatos de saída:
 1. Lista de atividades sequenciadas do projeto de pesquisa
 2. Lista de atividades sequenciadas de cada equipe
- Informações adicionais, Tabela B.32

Tabela B.32: Informações adicionais relacionadas à tarefa PI.2

Comentário:
O seqüenciamento de atividades pode ser realizado, por exemplo, usando o método do diagrama de precedência (MDP) ou o método do diagrama de setas (MDS). No MDP são usados retângulos, chamados de nós, para representar atividades. Os retângulos são conectados por setas, que indicam as dependências. De forma similar, o MDS é um método de construção de um diagrama de rede que usa setas para representar atividades, que são conectadas aos nós para indicar dependências (Project Management Institute, 2004).

PI.3 Estimativa de recursos das atividades

- Objetivo: determinar os recursos que serão necessários para cumprir as atividades do projeto.
- Artefatos de entrada: lista de atividades, disponibilidade de recursos
- Artefato de saída:
 1. Recursos para desenvolvimento do projeto
 - 1.1. Identificação de recursos para cada atividade (os detalhes e o nível de especificação das descrições dos recursos variam de acordo com as necessidades do projeto)

Planejamento de prazos

Identificador: PZ

Proposta: elaborar cronogramas para as equipes que participam do projeto de pesquisa e para projeto geral.

Resultados: como resultado da implementação bem sucedida deste processo espera-se que os prazos para a realização das atividades identificadas sejam estabelecidos

Tarefas: as tarefas relacionadas à atividade de planejamento de prazos foram identificadas a partir do PMBOK, da norma ISO/IEC 15504-5 e dos trabalhos de Maidantchik (1999) e de Humphrey (1999).

PZ.1 Elaboração de cronograma para cada equipe e para o projeto

- **Objetivos:** elaborar o cronograma individual para cada equipe, discutindo, se possível, a definição de prazos com os membros que cumprirão as atividades. Elaborar um cronograma geral para o projeto, considerando os cronogramas obtidos para cada equipe.
- **Artefatos de entrada:** lista de atividades do projeto de pesquisa, lista de atividades de cada equipe, recursos para desenvolvimento do projeto
- **Artefatos de saída:**
 1. Cronograma de cada equipe
 2. Cronograma do projeto (deve integrar os cronogramas de cada equipe e acrescentar as atividades relativas ao projeto)
- **Informações adicionais,** Tabela B.33

Tabela B.33: Informações adicionais relacionadas à tarefa PZ.1

Sugestão:
Gil (2002) sugere que uma representação bastante simples e prática seja utilizada na elaboração do cronograma de um projeto de pesquisa. O gráfico de Gantt pode ser utilizado, devendo-se criar uma tabela em que as atividades da pesquisa estejam representadas nas linhas e os períodos de tempo estejam representados nas colunas. O cronograma pode ser gerado utilizando-se um editor de texto, uma planilha ou uma ferramenta especializada. A ferramenta <i>GanttProject</i> , por exemplo, pode ser utilizada. Outras informações sobre a ferramenta podem ser encontradas em http://ganttproject.sourceforge.net/ .

Planejamento de custos

Identificador: PC

Proposta: estimar os custos para desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que o custo para o desenvolvimento do projeto de pesquisa seja estimado.

Tarefa: a tarefa relacionada à atividade de planejamento de custos foi identificada a partir do PMBOK, da norma ISO/IEC 15504-5 e do trabalho de Maidantchik (1999).

PC.1 Estimar custos

- **Objetivo:** estimar os recursos financeiros necessários para pagamento de bolsas, aquisição de materiais e equipamentos, viagens para reuniões e participações em eventos, dentre outros.
- **Artefatos de entrada:** identificação dos recursos humanos necessários, recursos para desenvolvimento do projeto, plano de pesquisa.

- Artefato de saída:
 1. Estimativa de custos
 - 1.1. Planilha relacionando atividades do projeto, recursos requeridos e a estimativa de custos

Planejamento de riscos

Identificador do processo: PS

Proposta: auxiliar a identificação e análise de possíveis riscos relacionados ao desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida deste processo espera-se que os riscos do desenvolvimento do projeto de pesquisa sejam identificados.

Tarefas: as tarefas relacionadas à atividade de planejamento de riscos foram identificadas a partir do PMBOK e da norma ISO/IEC 15504-5 e do trabalho de Sommerville (2003).

PS.1 Identificar riscos

- Objetivo: identificar e documentar os riscos que podem afetar o desenvolvimento do projeto de pesquisa.
- Artefatos de entrada: escopo do projeto, plano de pesquisa, lista de atividades, cronograma do projeto, estimativa de custos, recursos para desenvolvimento do projeto, artefatos gerados no processo de desenvolvimento.
- Artefato de saída: os riscos podem ser identificados realizando-se uma sessão de *brainstorming* com todos os membros da equipe para incentivar a discussão sobre riscos do projeto. Assim, devem ser identificados riscos que podem comprometer o desenvolvimento do projeto em relação a escopo, prazos, custo e qualidade. Cada coordenador local pode identificar os riscos envolvidos na realização das tarefas atribuídas à sua equipe ou os riscos podem ser identificados conjuntamente, em reunião com todas as equipes. Os riscos devem ser categorizados por tipo e devem ser descritos.
 1. Identificação de riscos
 - 1.1. Descrição do risco
 - 1.2. Categoria
 - 1.3. Área(s) afetada(s) do projeto (por exemplo, elementos da EAP)

PS.2 Analisar riscos

- Objetivo: avaliar a prioridade dos riscos identificados considerando a probabilidade de que ocorram, o impacto correspondente nos objetivos do projeto se os riscos realmente ocorrerem, bem como a tolerância a riscos considerando-se a qualidade exigida do projeto e as restrições de custo, cronograma e escopo.

- Artefato de entrada: identificação de riscos
- Artefato de saída:
 1. Análise de riscos: a avaliação de probabilidade de riscos investiga a probabilidade de cada risco específico ocorrer. A avaliação de impacto de riscos investiga o efeito potencial sobre um objetivo do projeto, como prazo, custo, escopo ou qualidade. De acordo com Sommerville (2003), na análise de riscos é feito um julgamento sobre a probabilidade de ocorrência de cada risco e o impacto associado. A probabilidade de um risco deve ser avaliada como alta, moderada ou baixa. O impacto de um risco deve ser avaliado como catastrófico, sério, tolerável ou insignificante. Na prática é importante ter informações sobre o projeto, o processo, o grupo de desenvolvimento e a organização para fazer análise de riscos. Deve-se notar que a probabilidade de ocorrência e as consequências de um risco podem mudar durante o desenvolvimento do projeto. Portanto, o artefato contendo informações sobre os riscos precisa ser constantemente atualizada (Tabela B.34).

Tabela B.34: Artefato sugerido para a atividade GM.2

Risco	Probabilidade	Efeitos (impacto)	Prioridade (alta, moderada, baixa)

PS.3 Planejar respostas aos riscos

- Objetivo: determinar ações que auxiliem a redução das ameaças aos objetivos e ao sucesso do projeto de pesquisa caso os riscos ocorram de fato.
- Artefatos de entrada: identificação de riscos, análise dos riscos
- Artefato de saída:
 1. Plano de resposta aos riscos: conforme indicado no documento do PMBOK (Project Management Institute, 2004), as respostas aos riscos precisam ser adequadas à importância do risco, econômicas ao enfrentar o desafio, rápidas, realistas dentro do contexto do projeto, acordadas por todas as equipes envolvidas e ser de responsabilidade de uma pessoa específica, que será responsável pela resposta ao risco. Geralmente consideram-se quatro estratégias para tratamento de risco: aceitar o risco, evitar o risco, suavizar o risco (reduzir sua probabilidade ou impacto) e decidir quais ações devem ser tomadas quando as demais estratégias falharem (contingência do risco).
 - 1.1. Risco identificado

- 1.2. Responsável pela resposta ao risco
- 1.3. Estratégias de resposta ao risco
- 1.4. Ações específicas para implementar a estratégia de resposta escolhida
- 1.5. Sintomas e sinais de alerta da ocorrência do risco

B.3.6 Processo de Divulgação

O processo de desenvolvimento é composto pelas atividades de implantação do processo e divulgação.

Implantação do processo

Identificador: IA

Proposta: identificar os mecanismos que serão utilizados para divulgar o projeto à comunidade externa e publicar os resultados às equipes de trabalho.

Resultados: como resultado da implementação bem sucedida deste processo espera-se que uma estratégia para divulgação do projeto e para a publicação de resultados seja definida e implantada.

Tarefa: a tarefa relacionadas à atividade de implantação do processo foi identificada a partir do trabalho de Boldyreff et al. (2004).

IA.1 Desenvolver política para divulgação

- **Objetivos:** determinar o(s) objetivo(s) da divulgação do projeto à comunidade externa e da publicação de resultados às equipes de trabalho.
- **Artefatos de entrada:** partes do projeto que serão divulgadas ou publicadas (artefatos, atividades, decisões, restrições, mudanças, etc).
- **Artefatos de saída:**
 1. Plano para divulgação do projeto (comunidade externa)
 - 1.1. Objetivos da divulgação
 - 1.2. Atividades de divulgação
 - 1.3. Meio(s) de publicação
 2. Plano para publicação dos resultados (equipes de trabalho)
 - 2.1. Informações sobre os artefatos e atividades do processo que serão publicadas
 - 2.2. Etapas do desenvolvimento do projeto em que tais informações serão disponibilizadas
 - 2.3. Meio de publicação
- **Informações adicionais,** Tabela B.35

Tabela B.35: Informações adicionais relacionadas à tarefa IA.1

Sugestão:
Boldyreff et al. (2004) discutiram brevemente os efeitos negativos observados pela equipe de desenvolvimento do projeto OSCAR devido à divulgação ineficiente. Tendo em vista as dificuldades experimentadas, os autores sugerem que, caso o objetivo da divulgação seja atrair usuários externos, o planejamento da divulgação do projeto inclua a apresentação do projeto em congressos, conferências, palestras e seminários; o desenvolvimento e a manutenção de um <i>site web</i> e a elaboração de um pacote de divulgação do projeto contendo material didático de apresentação do projeto (<i>slides</i> , apresentação multimídia, artigos e relatórios técnicos do projeto e outro tipo de documentação que seja interessante para potenciais usuários).

Divulgação

Identificador do processo: DI

Proposta: promover a divulgação do projeto de acordo com os objetivos de divulgação estabelecidos.

Resultados: como resultado da implementação bem sucedida deste processo espera-se que os resultados do projeto de pesquisa sejam divulgados à comunidade de interesse.

Tarefas: as tarefas relacionadas à atividade de implantação do processo foram identificadas a partir dos trabalhos de Boldyreff et al. (2004) e Lucrédio et al. (2006).

DI.1 Divulgar projeto de pesquisa

- Objetivo: disseminar os resultados do desenvolvimento do projeto de pesquisa
- Artefatos de entrada: plano para divulgação do projeto, artefatos que serão divulgados
- Artefatos de saída: são dependentes do plano de divulgação proposto. Podem ser, por exemplo, um *site web* atualizado ou um pacote de divulgação do projeto.

DI.2 Publicar resultados do projeto de pesquisa

- Objetivos: disponibilizar informações do projeto de pesquisa às equipes de trabalho.
- Artefatos de entrada: plano para publicação dos resultados, artefatos e atividades que serão publicados
- Artefato de saída: artefato disponível de acordo com o meio de publicação planejado.

B.3.7 Processo de Comunicação

O processo de comunicação é composto pelas atividades de implantação do processo e planejamento da comunicação.

Implantação do Processo

Identificador: IN

Proposta: desenvolver uma estratégia para a comunicação entre os membros que participam do desenvolvimento do projeto de pesquisa.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para comunicação seja definida e implantada.

Tarefa: a tarefa relacionada à atividade de implantação do processo foi identificada a partir do documento do PMBOK e do trabalho de Maidantchik (1999).

IN.1 Desenvolver política para comunicação

- **Objetivo:** identificar as metodologias e as tecnologias que deverão ser usadas para promover a comunicação entre as partes interessadas no desenvolvimento do projeto.
- **Artefatos de entrada:** recursos para desenvolvimento do projeto, relatório de infra-estrutura
- **Artefato de saída:** a possibilidade de comunicação entre os membros da equipe varia desde conversas breves até reuniões formais. Como resultado, diferentes metodologias e tecnologias podem ser utilizadas no decorrer do desenvolvimento do projeto e é comum que muitas alterações ocorram neste sentido a medida em que são utilizadas e avaliadas pelos membros.
 1. Plano para comunicação
 - 1.1. Metodologias e tecnologias de comunicação que cada equipe utilizará internamente
 - 1.2. Metodologias e tecnologias de comunicação que serão utilizadas entre as equipes

Planejamento da comunicação

Identificador: PC

Proposta: determinar as necessidades de informações e comunicação das equipes e identificar como a comunicação será realizada.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as necessidades de troca de informações entre as equipes, a notificação de decisões e o compartilhamento e a publicação de informações sejam planejadas.

Tarefas: as tarefas relacionadas à atividade de planejamento da comunicação foram identificadas a partir do documento do PMBOK e do trabalho de Maidantchik (1999).

PC.1 Estabelecer informações que serão transmitidas

- Objetivos: determinar as necessidades de informações das equipes que participam do desenvolvimento do projeto e garantir a transmissão de informações sobre o projeto de forma oportuna e adequada.
- Artefato de entrada: plano para comunicação
- Artefato de saída:
 1. Informações que serão transmitidas
 - 1.1. Informações que serão transmitidas internamente às equipes de trabalho (se necessário especificar formato, conteúdo e nível de detalhes)
 - 1.2. Informações que serão transmitidas entre equipes de trabalho (se necessário especificar formato, conteúdo e nível de detalhes)

PC.2 Planejar a comunicação

- Objetivo: definir diretrizes para que a comunicação ocorra
- Artefato de entrada: informações que serão armazenadas e transmitidas
- Artefato de saída: foi sugerido no documento do PMBOK que sejam registradas diretrizes que devam ser utilizadas para a realização de reuniões do projeto, notificação de decisões, etc. O plano de gerenciamento das comunicações pode ser formal ou informal, bem detalhado ou genérico, e deve se basear nas necessidades do projeto.
 1. Plano de comunicação:
 - 1.1. Itens de comunicação (por exemplo, *emails*, atas de reuniões)
 - 1.2. Objetivo (a razão da distribuição dos itens de comunicação)
 - 1.3. Frequência de distribuição das informações
 - 1.4. Datas de início/conclusão (prazo para a distribuição das informações)
 - 1.5. Meio físico para transmissão
 - 1.6. Responsabilidade (o(s) membro(s) da equipe encarregado(s) pela distribuição das informações)
- Informações adicionais, Tabela B.36

B.3.8 Processo de Coordenação

O processo de coordenação é composto pelas atividades de implantação do processo e execução e controle.

Tabela B.36: Informações adicionais relacionadas à tarefa PC.2

Comentário:
Considerando-se as dificuldades de comunicação que ocorrem quando um projeto é desenvolvido de forma distribuída, Paasvaara e Lassenius (2004) sugerem que sejam realizadas reuniões presenciais no início do projeto, principalmente se os membros não se conhecem. Segundo os autores, dentre outros benefícios, as reuniões prévias facilitam a comunicação eletrônica posterior.
Experiência:
Em relação à interação universidade-empresa, os resultados da experiência apresentada por Kornecki et al. (2003) indicaram que os investimentos dos membros em promover uma comunicação efetiva e em diversificar os mecanismos de comunicação utilizados tiveram resultados positivos. Foram realizadas visitas regulares na universidade e na empresa, houve o planejamento das reuniões, <i>emails</i> foram armazenados, atas indicando trabalhos realizados foram redigidas, contatos entre pesquisadores e representantes do setor industrial foram frequentes e teleconferências foram realizadas.

Implantação do Processo

Identificador: MR

Proposta: desenvolver uma estratégia para coordenação das equipes de trabalho.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que uma estratégia para coordenação de equipes seja definida e implantada.

Tarefa: a tarefa relacionadas à atividade de implantação do processo foi identificada a partir do trabalho de Maidantchik (1999).

MR.1 Preparar para a coordenação de equipes

- Objetivo: determinar as atividades de coordenação de equipes que serão cumpridas
- Artefatos de entrada: lista de atividades de cada equipe, cronograma de cada equipe
- Artefato de saída:
 1. Plano para coordenação
 - 1.1. Descrição de cada atividade de coordenação

Execução e controle

Identificador: EC

Proposta: coordenar equipes de trabalho.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que as equipes sejam coordenadas de acordo com o plano para coordenação proposto.

Tarefa: a tarefa relacionada à atividade de execução e controle foi identificada a partir do trabalho de Maidantchik (1999).

EC.1 Coordenar equipes de trabalho

- Objetivos: monitorar a execução das atividades cumpridas pelas equipes
- Artefato de entrada: plano para coordenação
- Artefato de saída:
 1. Registros de coordenação de equipes
 - 1.1. Análise do progresso do cumprimento de cada atividade
 - 1.2. Alterações em relação aos artefatos a serem gerados e justificativas
 - 1.3. Alterações em relação ao cronograma e justificativas

B.3.9 Processos para Estabelecimento de Parceria Universidade-Empresa

O processo de desenvolvimento é composto pela atividade de estabelecimento de parceria

Estabelecimento de parceria

Identificador: EA

Proposta: auxiliar a definição de elementos fundamentais relacionados à formação de parcerias, envolvendo identificação de parceiros e definição de contratos.

Resultados: como resultado da implementação bem sucedida desta atividade espera-se que sejam identificados parceiros para o desenvolvimento de projetos de pesquisa e sejam definidos contratos.

Tarefas: as tarefas relacionadas à atividade de estabelecimento de parceria englobam elementos considerados importantes nas políticas de iniciativa a transferência tecnológicas denominadas CRADAs (*Cooperative Research and Development Agreements*) (Rogers et al., 1998), dos Estados Unidos e modelo Fraunhofer, da Alemanha (Rombach, 2000).

EA.1 Identificar parceiros

- Objetivos: definir fatores que influenciam a identificação e a escolha de parceiros do setor acadêmico ou industrial e propor parcerias.
- Artefatos de entrada: documentos e relatórios descrevendo as condições políticas para a formação de parcerias.

- Artefato de saída: devem ser identificados os fatores (ou requisitos) que serão considerados para a escolha de parceiros. Estes fatores são dependentes, por exemplo, de condições impostas pelo regimento da universidade e das organizações que desejam se envolver em parcerias; de recursos disponíveis e dos focos, objetivos e metas dos membros do setor acadêmico e industrial.
 1. Fatores que influenciam a identificação de parceiros
 - 1.1. Requisitos para a escolha de parceiros
 - 1.2. Justificativas
- Informações adicionais, Tabela B.37

Tabela B.37: Informações adicionais relacionadas à tarefa EA.1

Experiências:
A partir da experiência na realização de parcerias entre universidade e indústria, Kornecki et al. (2003) indicaram um fator que consideraram importante para a identificação de parceiros: um dos membros da academia deve ser familiar com o domínio de aplicação e a cultura da organização.
De acordo com a experiência obtida no contexto da iniciativa CRADA, as empresas procuram parceiros do setor acadêmico conversando e trocando informações diretamente com pesquisadores, participando de conferências com o objetivo de conhecer os projetos em desenvolvimento e identificando a possibilidade de realização de parcerias considerando os resultados apresentados em artigos publicados.

EA.2 Estabelecer parcerias

- Objetivos: convidar membros para formação de parceria e manter registros sobre respostas obtidas e decisões tomadas.
- Artefatos de entrada: fatores que influenciam a identificação de parceiros
- Artefato de saída:
 1. Parcerias
 - 1.1. Nome do projeto ou sub-projeto para o qual a parceria será formada
 - 1.2. Breve descrição
 - 1.3. Entidade convidada e data
 - 1.4. Membro de contato (nome, email, endereço)
 - 1.5. Resposta obtida e data
 - 1.6. Decisão tomada
- Informações adicionais, Tabela B.38

EA.3 Definir e assinar contratos

Tabela B.38: Informações adicionais relacionadas à tarefa EA.2

Experiências:
Como lição aprendida durante o desenvolvimento do projeto Pro-Software, Jenkins (2004) enfatiza a importância na formação de um <i>steering committee</i> para guiar, de forma mais direta, o desenvolvimento, o monitoramento e o gerenciamento do projeto.
No modelo para transferência tecnológica do Japão, denominado Shuko, quando as parcerias são definidas, são realizados também intercâmbios entre membros do setor acadêmico e industrial por um determinado período. Quando eles retornam, levam com eles os processos aprendidos e as tecnologias desenvolvidas. Outro benefício observado é que a troca entre os membros auxilia na diminuição das diferenças culturais entre universidade e empresa, o que tem facilitado a formação de parcerias mais duradouras (Rogers et al., 1998).

- Objetivo: identificar necessidades de contratos formais entre as entidades participantes da parceria; elaborar e assinar contratos.
- Artefatos de entrada: parcerias
- Artefatos de saída: a elaboração de contratos é dependente dos objetivos das pessoas e das entidades envolvidas na parceria. Assim, uma tarefa que precisa ser realizada antes da definição de contratos é a identificação, de forma clara, dos objetivos de cada entidade que participa da parceria. Geralmente os modelos de contrato já existem nas instituições envolvidas. No entanto, é fundamental revê-los e adaptá-los de acordo com os interesses vigentes e com as experiências obtidas.
- Informações adicionais, Tabela B.39

As Tabelas B.40, B.41 e B.42 apresentam resumos dos relacionamentos entre os processos, as atividades e as tarefas do processo apresentado.

Tabela B.39: Informações adicionais relacionadas à tarefa EA.3

Experiência:
Kornecki et al. (2003), Macke et al. (1996) e Chang e Trubow (1990) indicaram alguns itens que julgam relevantes na elaboração de contratos (esses itens foram sugeridos como resultado de experiências pessoais na formação de parcerias): recursos provenientes da indústria que serão destinados à universidade, recursos provenientes da universidade que serão destinados à indústria, bolsas de estudos que serão oferecidas a pessoas envolvidas, períodos de trabalho para desenvolvimento de atividades referentes à parceria, definição de intercâmbio de profissionais, pesquisadores e estudantes na empresa e na universidade, criação de infra-estrutura na universidade de forma a suportar o desenvolvimento dos projetos que fazem parte do acordo estabelecido (equipamentos, software, prazos, <i>copyrights</i> , patentes, direito de posse, disponibilização de documentos), transferência de solução e tecnologia para o parceiro da indústria, possibilidade de publicação de resultados, tratamento a informações sigilosas e confidenciais, propriedade intelectual, detalhes de um plano de operação (detalhes do projeto a ser desenvolvido, métodos, pessoas, técnicas e artefatos a serem produzidos), definição de papéis e responsabilidades, código de conduta, direitos de comercialização, <i>royalties</i> , resultados esperados e indicação sobre avaliação dos parceiros envolvidos em relação aos resultados esperados.

Tabela B.40: Resumo dos identificadores definidos para os **processos fundamentais** do processo padrão de projetos de pesquisa em software

Processos	Atividades	Tarefas
Aquisição	AQ - Preparação para aquisição	AQ.1 - Definição dos requisitos para aquisição
		AQ.2 - Seleção de fornecedor
	AR - Aquisição de recursos	AR.1 - Preparação de contrato
AR.2 - Monitoramento de fornecedor		
AR.3 - Aceitação do recurso solicitado		
Iniciação	ET - Escolha do tema	ET.1 - Identificação dos temas de interesse
		ET.2 - Levantamento bibliográfico preliminar
		ET.3 - Formulação do problema
		ET.4 - Construção do repositório de terminologias
	AE - Análise de requisitos do sistema	AE.1 - Estabelecer os requisitos do sistema
		AE.2 - Analisar os requisitos do sistema
	PA - Projeto arquitetural do sistema	PA.1 - Projetar a arquitetura do sistema

Tabela B.40: continuação

Processos	Atividades	Tarefas
	EP - Elaboração de plano de pesquisa	EP.1 - Definição da metodologia científica
		EP.2 - Redação do plano de pesquisa
Desenvolvimento	DF - Definição de funcionalidades do protótipo	DF.1 - Especificar os requisitos do protótipo
	PR - Planejamento rápido	PR.1 - Planejamento
	PP - Projeto rápido	PP.1 - Elaboração de projeto simples
	DP - Desenvolvimento do protótipo	DP.1 - Implementação do código fonte
		DP.2 - Projetar procedimentos e casos de teste de unidade
		DP.3 - Executar testes de unidade
	AP - Avaliação do protótipo	AP.1 - Planejar testes funcionais
		AP.2 - Projetar testes funcionais
		AP.3 - Executar testes funcionais
	IR - Integração dos resultados da equipes	IR.1 - Planejar a integração dos resultados de cada equipe
IR.2 - Projetar testes de integração		
IR.3 - Executar testes de integração		
Operação	GU - Garantia do uso operacional do protótipo	GU.1 - Executar testes operacionais
	SU - Suporte ao usuário	SU.1 - Estabelecer e implantar um mecanismo de suporte
Manutenção	PM - Preparação para manutenção	PM.1 - Desenvolver estratégia de manutenção
	RM - Realizar manutenção no protótipo	RM.1 - Analisar o pedido de manutenção
		RM.2 - Documentar, implementar e testar a modificação
		RM.3 - Revisar e aceitar modificação
		RM.4 - Descontinuar o protótipo

Tabela B.41: Resumo dos identificadores definidos para os processos de apoio do processo padrão de projetos de pesquisa em software

Processos	Atividades	Tarefas
Documentação	IP - Implantação do processo	IP.1 - Desenvolver política para documentação
	PD - Planejamento da documentação	PD.1 - Gerar modelos para documentação

Tabela B.41: continuação

Processos	Atividades	Tarefas
	PO - Produção da documentação	PO.1 - Documentar o desenvolvimento do projeto de pesquisa
Gerenciamento de configuração	IO - Implantação do processo	IO.1 - Desenvolver política para gerenciamento de configuração
	II - Identificação de itens de configuração	II.1 - Estabelecer itens de configuração
	CC - Controle da configuração	CC.1 - Controlar alterações
		CC.2 - Controlar versões
AC - Avaliação da configuração	AC.1 - Avaliação do aspecto funcional de item de configuração	
Garantia de qualidade	IC - Implantação do processo	IC.1 - Desenvolver política para garantia de qualidade
	GQ - Garantia de qualidade de artefatos	GQ.1 - Garantir qualidade dos artefatos
	GA - Garantia de qualidade de processos	GA.1 - Garantir qualidade do processo
Verificação	IE - Implantação do processo	IE.1 - Desenvolver política para verificação
	VE - Verificação	VE.1 - Verificação dos artefatos
Validação	IS - Implantação do processo	IS.1 - Desenvolver política para validação
	VA - Validação	VA.1 - Validação dos artefatos
Revisão	IB - Implantação do processo	IB.1 - Desenvolver política para revisão
	RR - Realização de revisões	RR.1 - Realização de revisões gerenciais
		RR.2 - Realização de revisões técnicas
RR.3 - Realização de revisões de documentos científicos		
Resolução de problemas	ID - Implantação do processo	ID.1 - Desenvolver política de resolução de problemas
	RP - Resolução de problemas	RP.1 - Registro do problema e da solução
Revisão sistemática	IM - Implantação do processo	IM.1 - Desenvolver política para revisão sistemática
	RE - Realização da revisão sistemática	RE.1 - Planejar a revisão sistemática
		RE.2 - Executar a revisão sistemática
RE.3 - Registrar os resultados da revisão sistemática		
Preparação de documentos científicos	RD - Redação de documentos científicos	RD.1 - Definir os elementos estruturais do documento

Tabela B.41: continuação

Processos	Atividades	Tarefas
	FD - Formatação de documentos científicos	FD.1 - Apresentar documentos científicos de acordo com regras de formatação
Elaboração de módulos educacionais	EM - Elaboração de modelos para representação de conteúdo educacional	EM.1 - Determinar os aspectos específicos do domínio de conhecimento
		EM.2 - Refinar o modelo conceitual e definir elementos instrucionais
		EM.3 - Estabelecer ordem pedagógica para apresentação das informações
<i>Postmortem</i>	PE - Preparação e realização de análise <i>postmortem</i>	PE.1 - Planejamento de análise <i>postmortem</i>
		PE.2 - Coleta de dados
	FA - Finalização da análise <i>post-mortem</i>	FA.1 - Análise de dados
		FA.2 - Registro e publicação dos resultados
Transferência tecnológica	PT - Preparação para transferência tecnológica	PT.1 - Analisar a viabilidade da transferência tecnológica
		PT.2 - Realizar estudo observacional
	RS - Realização de estudos de caso e implantação do protótipo em ambiente industrial	RS.1 - Realizar estudos de caso em ambiente acadêmico
		RS.2 - Realizar estudos de caso em ambiente industrial
		RS.3 - Modificar o protótipo e implantar na indústria

Tabela B.42: Resumo dos identificadores definidos para os **processos organizacionais** do processo padrão de projetos de pesquisa em software

Processos	Atividades	Tarefas
Treinamento	IL - Implantação do processo	IL.1 - Desenvolver política para treinamento
	TR - Treinamento	TR.1 - Treinar equipes
Gerenciamento	GE - Gerenciamento de escopo	GE.1 - Gerenciamento de escopo
	GR - Gerenciamento de recursos humanos	GR.1 - Preparar para o recrutamento
		GR.2 - Preparar e tornar pública a chamada à participação no desenvolvimento de projetos de pesquisa

Tabela B.42: continuação

Processos	Atividades	Tarefas
		GR.3 - Avaliar perfil dos candidatos e realizar seleção
		GR.4 - Negociar e assinar termo de responsabilidade
		GR.5 - Registrar informações gerais sobre membros
	GC - Gerenciamento das capacidades das equipes	GC.1 - Determinar as capacidades de cada equipe
		GC.2 - Associar pessoas a papéis e responsabilidades
	GT - Gerenciamento de atividades	GT.1 - Gerenciamento de atividades das equipes
	GO - Gerenciamento de cronograma	GO.1 - Gerenciamento de cronograma das equipes
	GS - Gerenciamento de custos	GS.1 - Gerenciamento de custos das equipes
	GI - Gerenciamento de riscos	GI.1 - Gerenciamento de riscos das equipes
GN - Gerenciamento de requisitos	GN.1 - Gerenciamento de requisitos do projeto	
Infra-estrutura	IT - Implantação do processo	IT.1 - Definir recursos de infra-estrutura e planejar instalação
		IT.2 - Estabelecer políticas de uso da infra-estrutura
	IF - Implantação de infra-estrutura	IF.1 - Instalar e planejar a manutenção de recursos de infra-estrutura
Melhoria	DR - Determinação do processo	DR.1 - Descrever e manter a descrição de cada processo cumprido
	AO - Avaliação do processo	AO.1 - Definir plano de avaliação
		AO.2 - Realizar avaliação
	MP - Melhoria do processo	MP.1 - Estabelecer objetivos de melhoria
MP.2 - Melhorar processos		
Planejamento	DE - Definição de escopo do projeto	DE.1 - Identificar escopo inicial
		DE.2 - Criar EAP (Estrutura Analítica do Projeto)
	PI - Planejamento de atividades	PI.1 - Identificação de atividades de projeto
		PI.2 - Sequenciamento de atividades
		PI.3 - Estimativa de recursos das atividades
	PZ - Planejamento de prazos	PZ.1 - Elaboração de cronograma para cada equipe e para o projeto

Tabela B.42: continuação

Processos	Atividades	Tarefas
	PC - Planejamento de custos	PC.1 - Estimar custos
	PS - Planejamento de riscos	PS.1 - Identificar riscos
		PS.2 - Analisar riscos
		PS.3 - Planejar respostas aos riscos
Divulgação	IA - Implantação do processo	IA.1 - Desenvolver política para divulgação
	DI - Divulgação	DI.1 - Divulgar projeto de pesquisa
		DI.2 - Publicar resultados do projeto de pesquisa
Comunicação	IN - Implantação do processo	IN.1 - Desenvolver política para comunicação
	PC - Planejamento da comunicação	PC.1 - Estabelecer informações que serão transmitidas
		PC.2 - Planejar a comunicação
Coordenação	MR - Implantação do processo	MR.1 - Preparar para a coordenação de equipes
	EC - Execução e controle	EC.1 - Coordenar equipes de trabalho
Estabelecimento de parceria universidade-empresa	EA - Estabelecimento de parceria	EA.1 - Identificar parceiros
		EA.2 - Estabelecer parcerias
		EA.3 - Definir e assinar contratos