

COMPOOTIM: EM DIREÇÃO AO PLANEJAMENTO, ACOMPANHAMENTO E
OTIMIZAÇÃO DA COLABORAÇÃO NA DEFINIÇÃO DE PROCESSOS DE
SOFTWARE

Andréa Magalhães Magdaleno

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Cláudia Maria Lima Werner
Renata Mendes de Araujo

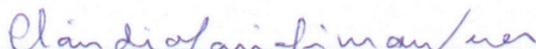
Rio de Janeiro
Junho de 2013

COMPOOTIM: EM DIREÇÃO AO PLANEJAMENTO, ACOMPANHAMENTO E
OTIMIZAÇÃO DA COLABORAÇÃO NA DEFINIÇÃO DE PROCESSOS DE
SOFTWARE

Andréa Magalhães Magdaleno

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM
CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:



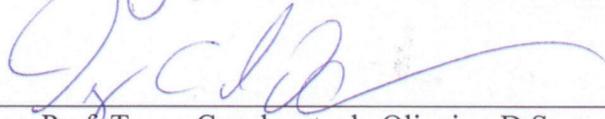
Prof.^a Cláudia Maria Lima Werner, D.Sc.



Prof.^a Renata Mendes de Araujo, D.Sc.



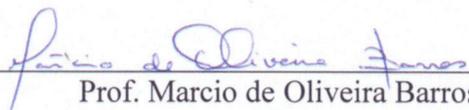
Prof.^a Ana Regina Cavalcanti da Rocha, D.Sc.



Prof. Toacy Cavalcante de Oliveira, D.Sc.



Prof. Rafael Prikladnicki, D.Sc.



Prof. Marcio de Oliveira Barros, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

JUNHO DE 2013

Magdaleno, Andréa Magalhães

COMPOOTIM: Em Direção ao Planejamento, Acompanhamento e Otimização da Colaboração na Definição de Processos de Software. – Rio de Janeiro: UFRJ/COPPE, 2013.

XVI, 253 p.: il.; 29,7 cm.

Orientadores: Cláudia Maria Lima Werner

Renata Mendes de Araujo

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2013.

Referencias Bibliográficas: p. 210-243.

1. Colaboração. 2. Processos de Software. 3. Otimização. 4. Redes Sociais. I. Werner, Cláudia Maria Lima *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos meus avós Avelino e Maria de Lourdes

AGRADECIMENTOS

À minha mãe e minha tia por todo amor e carinho que me dedicaram, me guiando ao longo da vida e construindo a base para esta realização. Por terem me acompanhado nos bons e maus momentos desta jornada.

À minha família querida e seus agregados especiais. Todos me incentivaram a chegar até aqui e torceram pelo meu sucesso. Também compreenderam minhas frequentes ausências nos últimos anos.

Ao Fabio Farzat, meu beça, por sempre se mostrar interessado em discutir e disponível para ajudar. Pela sua dedicação e envolvimento nos momentos decisivos deste trabalho. Por ter sido o meu melhor presente de 2012. Por me ensinar tantas coisas. Obrigada por cada momento juntos, por tudo que construímos até aqui. Agora poderemos sonhar novos planos e viver uma nova fase.

Agradeço à minha orientadora Claudia Werner, por ter acreditado em mim desde o início. Por saber me orientar me dando espaço para crescer e fazendo intervenções certas que sempre levavam para o caminho certo. Por não ter me deixado desanimar e pela confiança no meu potencial para construir tudo que inventava. Por ter se tornado, ao longo desse tempo, um exemplo de competência, inteligência, seriedade e bom senso. Por ainda estar disposta a continuar este trabalho e parceria no posdoc.

À amiga e co-orientadora Renata Araujo, pela amizade, disponibilidade, atenção e paciência dedicadas. Agradeço pelos inúmeros conselhos, pelas discussões e questionamentos, pelas revisões cuidadosas e pelo empenho em extrair sempre o melhor de mim. Não sei como, mas você conseguiu me aguentar em três orientações! Você formou a pesquisadora que sou hoje.

Ao meu “co-co”-orientador Marcio Barros, pela boa vontade para contribuir com este trabalho. Pelas ideias e sugestões. Pela paciência em sempre recomeçar. No limite, a sua dedicação e disponibilidade fizeram toda a diferença.

Ao Carlos Freud, por ter aberto as portas da empresa e se engajado em contribuir com este trabalho. Pelas horas e horas de reuniões e conversas dedicadas. Por se dispor a enfrentar os seus tramas e assistir a minha defesa.

Por ter me ajudado a acreditar que tudo que eu estava propondo realmente funcionava.

Às eternas amigas Juliana Jansen e Flavia Romero, que são um porto seguro e uma fonte de alegria. Obrigada pelas conversas filosóficas e pelo compartilhamento das angústias. Pelos esporros e colos nas horas certas.

Às meninas do NP2Tec, Claudia Cappelli, Flavia Santoro, Fernanda Baião, Vanessa Nunes, Hadeliane Iendrike e Ercilia Moreira, por terem me apoiado a embarcar nesta empreitada e terem dividido comigo as tarefas ingratas, me liberando para me dedicar a este trabalho.

A todos os meus companheiros de trabalho, que permitiram que eu investisse nesse sonho, me liberando para assistir as aulas e compreendendo quando eu precisava de tempo para escrever este texto.

A todos os meus amigos que colaboraram nos momentos de descontração, simplesmente com boa conversa ou companheirismo.

A todos os colegas e parceiros de pesquisa da COPPE e da UNIRIO: Taisa, Susie, Rodrigo, Marcelo, Marco, Danny, Rafael, Alice, Mozart e Caio. Todos participaram e contribuíram de alguma forma com o trabalho.

Aos professores Ana Regina Cavalcanti da Rocha, Toacy Cavalcante e Rafael Prikładnicki, que aceitaram prontamente o convite para participar da banca avaliadora e cujos comentários influenciaram diretamente a qualidade deste trabalho.

Agradeço também a todos que participaram das avaliações conduzidas neste trabalho e a todos os revisores dos artigos escritos, que contribuíram com críticas construtivas e apontaram direcionamentos.

À Dona Vera, por cuidar para que eu tivesse um ambiente confortável, aconchegante e sossegado para desenvolver a minha tese.

À Taisa e demais funcionários do PESC pela ajuda com os procedimentos administrativos, sempre que necessário.

Ao CNPq pelo apoio financeiro durante o doutorado e ao NP2Tec e PESC pelo auxílio financeiro para a apresentação de trabalhos em eventos.

Por fim, agradeço a todos que me apoiaram ao longo deste trabalho e que torceram pela realização deste sonho. Muito obrigada!

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

COMPOOTIM: EM DIREÇÃO AO PLANEJAMENTO, ACOMPANHAMENTO E
OTIMIZAÇÃO DA COLABORAÇÃO NA DEFINIÇÃO DE PROCESSOS DE
SOFTWARE

Andréa Magalhães Magdaleno

Junho/2013

Orientadores: Cláudia Maria Lima Werner
Renata Mendes de Araujo.

Programa: Engenharia de Sistemas e Computação

O desenvolvimento de software é tipicamente uma atividade colaborativa. No entanto, as organizações de software ainda têm dificuldades em planejar como a colaboração deve acontecer e acompanhar os resultados alcançados. Nesta tese, investiga-se o planejamento, composição, otimização e acompanhamento da colaboração em processos de software. Assim, é proposta a COMPOOTIM, que oferece apoio ao gerente de projeto na composição de um processo de software que otimize a colaboração com base no contexto do projeto. A COMPOOTIM inclui as seguintes soluções: i) uma estratégia de medição da colaboração em processos de software; ii) uma sistemática que define o passo-a-passo para a criação de uma linha de processos e para a composição dos processos a partir da linha; iii) ferramental de apoio com mecanismos de composição, otimização e visualização; e iv) uma ferramenta de visualização e análise de redes sociais (EvolTrack-SocialNetwork). Para avaliar a viabilidade desta tese, foram conduzidos dois estudos exploratórios no ambiente acadêmico e um estudo de caso no contexto real da indústria. Os resultados destes estudos fornecem indícios da viabilidade e aplicabilidade das soluções propostas.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

COMPOOTIM: TOWARDS PLANNING, MONITORING AND OPTIMIZATION OF
COLLABORATION IN SOFTWARE PROCESS DEFINITION

Andréa Magalhães Magdaleno

June/2013

Advisors: Cláudia Maria Lima Werner
Renata Mendes de Araujo.

Department: Systems and Computing Engineering

Software development is typically a collaborative activity. However, software organizations still have difficulties in planning how the collaboration should happen and monitoring the results. This thesis investigates the planning, composition, optimization and monitoring of collaboration in software processes. This work proposes COMPOOTIM, which offers support to the project manager in the composition of a software process that optimizes collaboration based on project context. The COMPOOTIM includes the following solutions: i) a measurement strategy of collaboration in software processes; ii) a systematic that defines the step-by-step for the process line creation and the processes composition from the line; iii) tool support with composition, optimization and visualization mechanisms; and iv) a social network visualization and analysis tool (EvolTrack-SocialNetwork). To evaluate the feasibility of this thesis, two exploratory studies were conducted in the academic environment and a case study in the real context of industry. The results provide initial evidence of the feasibility and applicability of the proposed solutions.

SUMÁRIO

1. INTRODUÇÃO	1
1.1. Motivação.....	1
1.2. Caracterização do Problema.....	2
1.3. Enfoque de Solução.....	4
1.4. Contexto.....	6
1.5. Objetivo.....	7
1.6. Metodologia de Pesquisa.....	8
1.7. Organização do Trabalho.....	11
2. COLABORAÇÃO.....	13
2.1. Definição.....	15
2.2. Aspectos de Apoio à Colaboração.....	16
2.2.1. <i>Comunicação</i>	17
2.2.2. <i>Coordenação</i>	18
2.2.3. <i>Memória de grupo</i>	19
2.2.4. <i>Percepção</i>	19
2.3. Modelos de Maturidade em Colaboração.....	20
2.4. Redes Sociais.....	23
2.4.1. <i>Propriedades Topológicas</i>	25
2.4.2. <i>Metodologia para estudo de redes sociais</i>	26
2.4.3. <i>Análise de redes sociais</i>	27
2.4.4. <i>Análises de redes sociais no desenvolvimento de software</i>	31
2.4.5. <i>Ferramentas para o estudo de redes sociais</i>	33
2.5. Considerações Finais.....	37
3. REUTILIZAÇÃO DE PROCESSOS DE SOFTWARE.....	38
3.1. Diversidade de Processos de Software.....	40
3.1.1. <i>Colaboração como um Aspecto da Diversidade de Processos de Software</i> 41	
3.2. Estratégias de Reutilização de Processos de Software.....	46
3.3. Composição de Processos de Software.....	47
3.3.1. <i>Componentes de Processos</i>	47
3.3.2. <i>Linha de Processos de Software (LPS)</i>	49
3.4. Reutilização de Processos de acordo com Contexto.....	54

3.5.	Considerações Finais.....	57
4.	PROJETO CDSOFT.....	59
4.1.	Visão Geral do Projeto CDSOft	60
4.2.	Engenharia de Linha de Processos de Software (ELPS).....	62
4.2.1.	<i>Meta-Modelo e Notação.....</i>	62
4.2.2.	<i>Método</i>	69
4.2.3.	<i>Ferramental de apoio.....</i>	69
4.3.	Gestão de Contexto (GC)	70
4.3.1.	<i>Estrutura de representação de contexto.....</i>	71
4.3.2.	<i>Modelo de Contexto para Processos de Software.....</i>	72
4.4.	Linha de Processos Baseada em Contexto (LPBC)	77
4.4.1.	<i>Engenharia de Domínio de Processo (EDP).....</i>	78
4.4.2.	<i>Engenharia de Aplicação do Processo (EAP).....</i>	80
4.5.	Composição e Otimização de Processos.....	82
4.6.	COMPOOTIM no Projeto CDSOft.....	84
4.7.	Considerações Finais.....	86
5.	COMPOOTIM	87
5.1.	Planejamento da Colaboração	88
5.1.1.	<i>Modelo de Maturidade em Colaboração (CollabMM)</i>	89
5.1.2.	<i>Estratégia de Medição da Colaboração</i>	93
5.2.	Composição da Colaboração	98
5.2.1.	<i>Sistemática.....</i>	99
5.2.2.	<i>Ferramental de Apoio.....</i>	105
5.3.	Otimização da Colaboração	106
5.3.1.	<i>Mecanismo de Otimização.....</i>	106
5.3.2.	<i>Modelagem do Problema de Composição do Processo</i>	110
5.3.3.	<i>Algoritmos de Otimização</i>	116
5.4.	Acompanhamento da Colaboração.....	117
5.4.1.	<i>Características de colaboração em redes sociais de desenvolvimento de software</i>	117
5.4.2.	<i>Ferramenta EvolTrack-SocialNetwork</i>	120
5.5.	Apoio Computacional	131
5.5.1.	<i>Funcionalidades do Mecanismo de Composição.....</i>	132
5.5.2.	<i>Funcionalidades do Mecanismo de Otimização.....</i>	138
5.5.3.	<i>Funcionalidades do Mecanismo de Visualização</i>	138

5.6.	Considerações Finais.....	139
6.	ESTUDOS EXPERIMENTAIS	141
6.1.	Estudo Exploratório – Medição da Colaboração	142
6.2.	Estudo Exploratório – EvolTrack-SocialNetwork.....	144
6.2.1.	<i>Planejamento do Estudo Exploratório.....</i>	<i>144</i>
6.2.2.	<i>Execução do Estudo Exploratório.....</i>	<i>149</i>
6.2.3.	<i>Análise dos Resultados do Estudo Exploratório</i>	<i>150</i>
6.3.	Estudo de Caso – Etapa 1 – Avaliação da Criação da Linha de Processos.....	154
6.3.1.	<i>Planejamento</i>	<i>154</i>
6.3.2.	<i>Execução</i>	<i>159</i>
6.3.3.	<i>Análise dos Resultados.....</i>	<i>172</i>
6.3.4.	<i>Conclusões e Limitações</i>	<i>177</i>
6.4.	Estudo de Caso – Etapa 2 – Avaliação da Composição dos Processos	178
6.4.1.	<i>Planejamento</i>	<i>178</i>
6.4.2.	<i>Execução</i>	<i>182</i>
6.4.3.	<i>Análise dos Resultados.....</i>	<i>189</i>
6.4.4.	<i>Conclusões e Limitações</i>	<i>192</i>
6.5.	Avaliação da Otimização da Colaboração	193
6.6.	Considerações Finais.....	196
7.	CONCLUSÃO	198
7.1.	Epílogo.....	198
7.2.	Contribuições.....	200
7.2.1.	<i>Principais Contribuições.....</i>	<i>200</i>
7.2.2.	<i>Contribuições Secundárias</i>	<i>201</i>
7.2.3.	<i>Resultados em Publicações e Orientações</i>	<i>201</i>
7.3.	Limitações.....	204
7.4.	Perspectivas Futuras	205
	REFERÊNCIAS BIBLIOGRÁFICAS	210
	APÊNDICES.....	244
	Apêndice I – Termo de Consentimento Livre Esclarecido do Estudo Exploratório da EvolTrack-SocialNetwork.....	244

Apêndice II – Formulário de Caracterização do Participante do Estudo Exploratório da EvoITrack-SocialNetwork.....	246
Apêndice III – Formulário para Realização do Estudo Exploratório da EvoITrack-SocialNetwork.....	248
Apêndice IV – Questionário de Avaliação do Estudo Exploratório da EvoITrack-SocialNetwork.....	251

ÍNDICE DE FIGURAS

Figura 1.1 – Metodologia de Pesquisa	8
Figura 2.1 – Esquema geral dos aspectos de apoio à colaboração (ARAUJO e BORGES, 2007)	17
Figura 2.2 – Exemplos de redes (BARABASI e BONABEAU, 2003, WWF, 2003)	24
Figura 2.3 – Exemplo de rede para estudo das propriedades topológicas	25
Figura 2.4 – Exemplo de centralidade de grau	29
Figura 2.5 – Exemplo de centralidade de proximidade	30
Figura 2.6 – Exemplo de centralidade de intermediação	30
Figura 2.7 – Exemplo de densidade da rede (WWF, 2003)	31
Figura 2.8 – Rede de dependências técnicas no código (TRAINER ET AL., 2005).....	32
Figura 2.9 – Rede sócio-técnica (TRAINER <i>et al.</i> , 2005).....	33
Figura 2.10 – Rede de dependências sociais (TRAINER <i>et al.</i> , 2005).....	33
Figura 2.11 – Interface da ferramenta OSSNetwork (BALIEIRO <i>et al.</i> , 2008).....	36
Figura 3.1 – Linha de Processos de Aquisição de Software (NUNES <i>et al.</i> , 2010a) ...	52
Figura 3.2 – Abordagem <i>bottom-up</i> para linha de processos (WASHIZAKI, 2006)	53
Figura 3.3 – Visão Geral da Abordagem OSPTA (PEREIRA, 2012).....	56
Figura 3.4 – Abordagem para gerenciamento dinâmico de ativos de processos de software (SANTOS, 2009).....	57
Figura 4.1 – Visão geral da solução para composição de processos no Projeto CDSOft	60
Figura 4.2 – Elementos da ELPS (TEIXEIRA, 2011).....	63
Figura 4.3 – Exemplo de modelo de características com variabilidade.....	64
Figura 4.4 – Exemplo parcial de modelo de características de processo na notação <i>Odyssey-ProcessFEX</i>	68
Figura 4.5 – Visão geral da abordagem UbiFEX	71
Figura 4.6 – Dimensões de contexto no desenvolvimento de software (ARAUJO <i>et al.</i> , 2004).....	73
Figura 4.7 – Exemplo parcial de dimensões e informações de contexto no Odyssey .	75
Figura 4.8 – Método da LPBC	78
Figura 4.9 – COMPOOTIM no contexto do Projeto CDSOft	84
Figura 4.10 – Mapa de trabalhos de pesquisa ramificados a partir da COMPOOTIM .	85
Figura 5.1 – Visão Geral das Etapas e Soluções na COMPOOTIM	87
Figura 5.2 – Níveis e Práticas do CollabMM	90
Figura 5.3 – Passos e artefatos da fase de Análise de domínio de processos	100
Figura 5.4 – Passos e artefatos da fase de Projeto de domínio de processos	101

Figura 5.5 – Passos e artefatos da fase de Análise de aplicação de processos.....	103
Figura 5.6 – Passos e artefatos da fase de Projeto de aplicação de processos.....	104
Figura 5.7 – Passos e artefatos da etapa de Otimização.....	105
Figura 5.8 – Mecanismos da COMPOOTIM.....	105
Figura 5.9 – Resumo das etapas do PCPP.....	110
Figura 5.10 – Exemplos de redes sociais – Planejada (a) e Atual (b).....	119
Figura 5.11 – Visão geral da arquitetura da EvoTrack-SocialNetwork	123
Figura 5.12 – Visão geral da interface da EvoTrack-SocialNetwork usando o projeto Floggy	125
Figura 5.13 – Exemplo de rede técnica do projeto Floggy	126
Figura 5.14 – Exemplo de rede sócio-técnica particionada do projeto Floggy	127
Figura 5.15 – Painel de análise de redes sociais	129
Figura 5.16 – Visão Geral da Infraestrutura Computacional	131
Figura 5.17 – Tela inicial da COMPOOTIM	132
Figura 5.18 – Características de processo na COMPOOTIM	133
Figura 5.19 – Componentes de Processo na COMPOOTIM.....	134
Figura 5.20 – Cálculo do Potencial de Colaboração do Componente na COMPOOTIM	134
Figura 5.21 – Regras de Contexto na COMPOOTIM.....	135
Figura 5.22 – Informações de Contexto do Projeto na COMPOOTIM	135
Figura 5.23 – Situações e Regras de Contexto do Projeto determinadas pela COMPOOTIM	136
Figura 5.24 – Filtro das Características pelas Regras de Contexto e Composição aplicadas pela COMPOOTIM	137
Figura 5.25 – Filtro dos Componentes pelas Características e Regras de Composição aplicadas pela COMPOOTIM	137
Figura 5.26 – Início da otimização da colaboração na COMPOOTIM.....	138
Figura 5.27 – Visualização das Opções de Processos na COMPOOTIM.....	139
Figura 6.1 – Rede social de dependência do projeto TED.....	151
Figura 6.2 – Análises dos defeitos encontrados.....	161
Figura 6.3 – Fase Planejamento - Modelo de características no Odyssey	168
Figura 6.4 – Dimensões e Informações de Contexto modeladas no Odyssey.....	169
Figura 6.5 – Situações e Regras de Contexto do Projeto B na COMPOOTIM	185
Figura 6.6 – Filtro de Características do Projeto B na COMPOOTIM	186
Figura 6.7 – Filtro de Componentes do Projeto B na COMPOOTIM.....	187
Figura 6.8 – Resultado da Otimização do Projeto B na COMPOOTIM	188
Figura 6.9 – <i>Box-plot</i> do Projeto A.....	195

ÍNDICE DE TABELAS

Tabela 2.1 – Tipos das ferramentas de redes sociais	35
Tabela 3.1 – Resumo dos aspectos de colaboração nos modelos de desenvolvimento de software	45
Tabela 3.2 – Estrutura para definição de componentes de processo de software (NUNES, 2011a).....	49
Tabela 4.1 – Categorias das características na notação <i>OdysseyProcess-FEX</i> (TEIXEIRA, 2011).....	66
Tabela 4.2 – Estrutura para definição de componentes de processo de software.....	67
Tabela 4.3 – Características de contexto (FERNANDES, 2008).....	72
Tabela 4.4 – Exemplos de informações de contexto para processos de software.....	74
Tabela 4.5 – Exemplos de situações de contexto de equipe	76
Tabela 4.6 – Exemplos de regras de contexto	77
Tabela 5.1 – Classificação das Práticas de Colaboração de acordo com os Aspectos de Apoio à Colaboração e os Níveis de Maturidade	93
Tabela 5.2 – Aspectos de Apoio e Práticas de Colaboração	94
Tabela 5.3 – Componente de Processo Realizar Reunião Diária de Acompanhamento	95
Tabela 5.4 – Exemplo de Avaliação do Potencial de Colaboração do Componente de Processo.....	95
Tabela 5.5 – Exemplos de Componentes de Processo e seus respectivos PCs	97
Tabela 5.6 – Cálculo do nível de colaboração do processo em cada estratégia	98
Tabela 5.7 – Estrutura para definição de LPS.....	99
Tabela 5.8 – Características de coordenação em redes sociais de desenvolvimento de software (SANTOS <i>et al.</i> , 2010)	118
Tabela 5.9 – Requisitos para ferramentas de redes sociais.....	122
Tabela 5.10 – Tabela ferramentas x requisitos com a inclusão da <i>EvoTrack-SocialNetwork</i>	130
Tabela 6.1 – Descrição dos Componentes de Processo.....	142
Tabela 6.2 – Exemplos de Componentes de Processo e seus respectivos PCs	143
Tabela 6.3 – Tarefas propostas para o estudo.....	145
Tabela 6.4 – Caracterização dos Participantes	150
Tabela 6.5 – Resultados das respostas às tarefas do estudo.....	150
Tabela 6.6 – Eficiência de tempo (em minutos)	152
Tabela 6.7 – <i>Feedback</i> dos participantes.....	152
Tabela 6.8 – Caracterização do Participante.....	160

Tabela 6.9 – Conhecimento do Processo de Software da Organização	160
Tabela 6.10 – Revisão por Pares	162
Tabela 6.11 – Totalização das características da LPS.....	165
Tabela 6.12 – Regras de composição de características.....	166
Tabela 6.13 – Informações de Contexto.....	167
Tabela 6.14 – Totalização dos componentes da LPS	170
Tabela 6.15 – Regras de Composição de Componentes.....	170
Tabela 6.16 – Situações de Contexto.....	171
Tabela 6.17 – Regras de Contexto.....	172
Tabela 6.18 – Processos da Organização.....	173
Tabela 6.19 – Linha de Processos Resultante	173
Tabela 6.20 – Dificuldades encontradas	174
Tabela 6.21 – Necessidade das etapas e artefatos	174
Tabela 6.22 – Vantagens, Desafios e Melhorias	176
Tabela 6.23 – Caracterização do Projeto A.....	182
Tabela 6.24 – Caracterização do Projeto B.....	183
Tabela 6.25 – Opções de Processo do Projeto B.....	189
Tabela 6.26 – Comentários sobre as avaliações dos processos	189
Tabela 6.27 – Viabilidade	190
Tabela 6.28 – Vantagens, Desafios e Melhorias	191
Tabela 6.29 – Dados dos Projetos	194
Tabela 6.30 – Valores de <i>Fitness</i> dos algoritmos para o Projeto A.....	195
Tabela 6.31 – Valores de Tempo de Execução do Projeto A (em ms).....	196
Tabela 7.1 – Resumo das avaliações realizadas	199

1. Introdução

Este capítulo descreve a motivação e caracteriza o problema estudado. Em busca de solução para o problema apresentado, é discutido o enfoque de solução adotado neste trabalho. Em seguida, são declarados o objetivo geral e os objetivos específicos desta tese. Por fim, é resumida a forma como os capítulos desta tese foram organizados.

1.1. Motivação

Nas últimas décadas assistimos à globalização dos mercados. O atual cenário mundial oferece novas oportunidades de negócios às empresas, mas também apresenta grandes desafios. Neste ambiente competitivo, as organizações precisam de flexibilidade e agilidade para responder às demandas dos clientes, oferecendo rapidamente produtos e serviços de qualidade e personalizados.

Neste contexto, o trabalho em grupo se tornou uma importante estratégia de negócios e vem sendo demandado como instrumento para vencer os desafios do novo mundo globalizado (DEN HENGST *et al.*, 2006, MISTRİK *et al.*, 2010, SARMENTO, 2002, SCHOLTES *et al.*, 2003, SHIRKY, 2003, SOUZA *et al.*, 2011, SWENSON *et al.*, 2011, TELLERIA *et al.*, 2002). Assim, as organizações estão cada vez mais interessadas na **colaboração** para fins de qualidade, produtividade e compartilhamento de conhecimento (ALONSO *et al.*, 2010).

Em particular, o **desenvolvimento de software** é um processo complexo que envolve a colaboração de diversas pessoas durante um período de tempo para atingir um objetivo comum (BARTHELMESS, 2003, CUGOLA e GHEZZI, 1998). As pessoas exercem um papel preponderante para o sucesso de projetos de software. Elas participam de todo o ciclo de vida em diferentes papéis (clientes, usuários, gerentes, analistas, desenvolvedores e etc.), executam as atividades, compartilham artefatos e conhecimento (ALTMANN e POMBERGER, 1999, DEMARCO e LISTER, 1999, MISTRİK *et al.*, 2010, YILDRIM, 2006).

Portanto, o desenvolvimento de software é um típico exemplo de trabalho colaborativo (ALTMANN e POMBERGER, 1999, DEMARCO e LISTER, 1999, MISTRİK *et al.*, 2010, OMORONYIA *et al.*, 2010, SOUZA *et al.*, 2011, YILDRIM, 2006). Whitehead *et al.* (2010) chegam a afirmar que “a

colaboração é pervasiva por toda a Engenharia de Software”, pois quase todos os projetos de software não triviais requerem o esforço e talento de múltiplas pessoas trabalhando em conjunto.

Apesar dos seus potenciais benefícios, alcançar uma colaboração efetiva ainda é um desafio (ALONSO *et al.*, 2010, BORRELLI *et al.*, 1995, VREEDE e BRIGGS, 2005). As organizações ainda têm dificuldades em planejar e explicitar como querem que a colaboração aconteça no seu dia-a-dia de trabalho, tornar este planejamento da colaboração visível para todos os seus membros e acompanhar os resultados alcançados.

Na Engenharia de Software (ES), ainda se discute quais práticas, processos e ferramentas são capazes de estimular a colaboração e acompanhar sua execução (MISTRİK *et al.*, 2010). Tal fato é reforçado pelas pesquisas sobre desenvolvimento distribuído de software (CATALDO e HERBSLEB, 2008, HERBSLEB *et al.*, 2005, OMORONYIA *et al.*, 2010), métodos ágeis (COCKBURN e HIGHSMITH, 2001, COCKBURN, 2001), software livre (RAYMOND, 2001) e ecossistemas (BOSCH e BOSCH-SIJTSEMA, 2010), onde o foco nas pessoas (nos seus talentos, habilidades e conhecimentos) e a preocupação com a colaboração e comunicação aparecem de forma recorrente.

Desta forma, nota-se a necessidade de planejar, otimizar e acompanhar a colaboração no desenvolvimento de software. Compreender como as pessoas se organizam e interagem umas com as outras enquanto desenvolvem software pode ser importante para que as organizações alcancem os resultados pretendidos com a colaboração (VALETTO *et al.*, 2007).

1.2. Caracterização do Problema

Para projetar a colaboração em processos de software, deve-se considerar os desafios enfrentados com a diversidade. A diversidade de organizações e projetos faz com que os contextos onde os processos serão utilizados variem muito. A diversidade de modelos de desenvolvimento tem levado a uma busca sobre formas de conciliar as particularidades de cada modelo para a definição de processos de software mais eficazes. Por fim, um processo não pode ser

definido sem levar em consideração as pessoas que vão executá-lo (como os funcionários da organização) ou interagir com ele (como os clientes e fornecedores). Porém, a diversidade de pessoas envolvidas (com diferentes motivações, experiências, conhecimentos e etc.) e as interações sociais entre elas, acrescentam mais um desafio à tarefa de definição de processos de software.

Para lidar com toda esta diversidade, há uma crescente necessidade por parte da indústria de software pela rápida e efetiva definição de processos de software para os projetos (ALEIXO *et al.*, 2010, HANSSON *et al.*, 2006, PATEL *et al.*, 2006). Isto envolve a adequação ao **contexto** dos projetos e o **reúso** de experiências passadas na definição de processos de software com o objetivo de aumentar a produtividade durante a realização de tal atividade.

Como forma de promover a reutilização do conhecimento relacionado a processos de software, pode-se investir na **composição** de processos. Ao compor um processo com base em unidades menores e reutilizáveis, é possível introduzir a colaboração como o aspecto de interesse que se deseja otimizar durante esta combinação das unidades que irão formar o processo.

Porém, a composição de processos não é uma tarefa simples (TERNITE, 2009), devido à complexidade do desenvolvimento e à diversidade mencionada anteriormente. Esta tarefa envolve o conhecimento de muitos aspectos da Engenharia de Software e requer a harmonização de muitos fatores do contexto da equipe, do projeto ou da organização (BARRETO, 2007). Compreender esses fatores e como eles poderão afetar o processo de software é um desafio para o gerente de projeto (XU e RAMESH, 2008), principalmente para aqueles com menos experiência.

Como resultado, o processo definido pode não ser a melhor alternativa para o projeto em questão. De fato, a quantidade e diversidade de informações envolvidas fazem com que a composição de processos seja considerada uma tarefa complexa, demorada e propensa a erros quando realizada sem o devido apoio computacional.

Como o orçamento do projeto, o prazo de desenvolvimento e a qualidade do produto dependem diretamente da qualidade do processo de software, a composição de processos precisa ser feita corretamente para evitar que as seguintes possíveis consequências negativas se materializem: um

processo de software ruim pode envolver atividades desnecessárias que levam a perda de tempo e dinheiro; e a omissão de atividades necessárias pode prejudicar a qualidade do produto (PEDREIRA *et al.*, 2007).

Neste cenário, entende-se que a colaboração seja um aspecto passível de ser planejado e acompanhado nos processos compostos para distintos projetos de desenvolvimento de software.

O planejamento da colaboração deve permitir às organizações de software estabelecer objetivos e definir ações para atingir níveis mais altos de colaboração. O acompanhamento da colaboração diz respeito a monitorar qual é o nível real de colaboração no projeto e avaliar os resultados alcançados com a colaboração no processo, identificando se o nível planejado de colaboração ainda se mantém o mesmo ou se deve ser replanejado.

Assim, chegamos ao problema de pesquisa desta tese:

Como planejar, acompanhar e otimizar a colaboração através de uma sistemática de reutilização e composição de processos de software, e o ferramental de apoio relacionado, capaz de sugerir um processo adequado ao contexto de um determinado projeto?

1.3. Enfoque de Solução

Visando solucionar o problema de pesquisa apresentado na seção anterior, neste trabalho de pesquisa busca-se estabelecer uma solução que permita planejar, compor, otimizar e acompanhar a colaboração em processos de software. Assim, foi construída uma estratégia de composição de processos de software baseada em unidades reutilizáveis – como os componentes e as Linhas de Processo de Software (LPS)¹ (ALEIXO *et al.*, 2010, BARRETO, 2007, JAUFMAN e MUNCH, 2005, NUNES *et al.*, 2010b, ROMBACH, 2006, TERNITE, 2009, WASHIZAKI, 2006).

Segundo Costa (2010), definir mecanismos para auxiliar a composição de processos pode reduzir esforços de modelagem, garantir a qualidade dos

¹ Um conjunto de elementos de processos que compartilham características comuns e variáveis dentro de um domínio específico e são desenvolvidas a partir de artefatos que podem ser reutilizados e combinados entre si, segundo regras de composição e recorte, para compor e adaptar processos (MAGDALENO *et al.*, 2012a, NUNES *et al.*, 2010b, TEIXEIRA, 2011).

processos e impedir que erros já resolvidos em projetos passados tornem a trazer problemas em novos projetos. Com o propósito de facilitar a composição de processos, é possível fornecer apoio ao gerente de projeto na execução desta atividade, uma vez que alguns dos passos para a solução do problema podem ser automatizados (COSTA, 2010, PARK *et al.*, 2006), possivelmente diminuindo o esforço necessário para sua execução e melhorando os resultados obtidos.

Este apoio deve fornecer ao gerente de projeto um melhor embasamento para a tomada de decisões durante a composição de processos, uma vez que possibilita que seja considerado um universo maior de alternativas. O apoio sugerido diz respeito a propor ao gerente de projeto opções de processo que satisfaçam as restrições de contexto do projeto.

Neste cenário, é proposta a **COMPOOTIM**, voltada para o planejamento, composição, otimização e acompanhamento da colaboração nos processos compostos para os projetos de software. A COMPOOTIM construiu ou aplicou diferentes soluções para apoiar cada uma destas etapas, conforme discutido a seguir.

Na etapa de planejamento, para definir os objetivos e o nível de colaboração desejado ou necessário para o processo que será composto para o projeto, foi utilizado o Modelo de Maturidade em Colaboração (*Collaboration Maturity Model - CollabMM*) (MAGDALENO *et al.*, 2007a, 2008a, 2008b, 2009a) proposto anteriormente. Ele funciona como uma referência, definindo os níveis de colaboração em processos e resumindo as principais características de cada nível. Este modelo também organiza as principais práticas de colaboração que podem ser aplicadas aos processos para garantir/estimular a colaboração.

A partir do CollabMM, foi estabelecida uma **estratégia de medição da colaboração** em processos de software. Esta estratégia define o cálculo do potencial de colaboração de cada componente de processo, bem como sugere estratégias para calcular a colaboração do processo como um todo dada a combinação do potencial de colaboração dos seus componentes de processo.

Para auxiliar o gerente de projeto na composição de um processo que otimize o aspecto da colaboração, foi construída na COMPOOTIM tanto uma **sistemática** e um **ferramental de apoio**. A sistemática da COMPOOTIM

define o passo-a-passo para a criação da uma LPS e para a composição dos processos a partir da LPS. O ferramental de apoio construído inclui mecanismos de composição, otimização e visualização.

Para o acompanhamento da colaboração, a análise de **redes sociais**² (BARABASI, 2003) se apresenta como um caminho promissor, pois existem diversos trabalhos (CATALDO e HERBSLEB, 2008, CATALDO *et al.*, 2008, GAO *et al.*, 2003, GOTO *et al.*, 2008, HERBSLEB *et al.*, 2000, JIN XU *et al.*, 2005, LOPEZ-FERNANDEZ *et al.*, 2004, MADEY *et al.*, 2002) que apontam para o potencial das redes sociais em explicitar como a colaboração acontece dentro de um grupo. Desta forma, o entendimento das redes sociais envolvidas nos projetos de desenvolvimento ajuda a compreender e acompanhar o nível de colaboração do projeto.

A partir da combinação dos níveis do CollabMM com as propriedades de análise das redes sociais, foram propostas por (SANTOS *et al.*, 2010) as **características de colaboração** especificamente voltadas para o aspecto da coordenação. A ideia principal é que as informações obtidas através da análise de redes sociais podem ser associadas aos níveis de maturidade em colaboração do CollabMM, para estabelecer a característica de colaboração de uma determinada rede social.

Por fim, foi desenvolvida a ferramenta para mineração, visualização e análise das redes sociais - **EvoITrack-SocialNetwork** (VAHIA *et al.*, 2011) - para oferecer à equipe de desenvolvimento e ao gerente de projeto uma maior percepção sobre como a colaboração acontece em um grupo.

1.4. Contexto

Este trabalho de pesquisa está contextualizado no Projeto CDSOFT (WERNER *et al.*, 2011a), que visa construir soluções para o apoio à composição de processos de software de forma sistemática e dinâmica.

O Projeto CDSOFT reúne o Grupo de Reutilização de Software do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ

² As redes sociais consistem em um “conjunto finito de atores e as relações definidas entre eles” (WASSERMAN e FAUST, 1994). Em uma rede social, os nós representam os atores e as arestas correspondem aos possíveis relacionamentos entre eles. A semântica do relacionamento depende da análise que se deseja conduzir nesta rede.

e o Programa de Pós-Graduação em Informática da UNIRIO. Em parceria, estes dois grupos estão desenvolvendo atualmente o projeto com um grupo formado por: docentes de ambas as instituições; três alunos de doutorado, sendo um deles a autora desta tese; e alguns alunos de mestrado e graduação, alocados em trabalhos mais específicos.

As estratégias de solução aplicadas no Projeto CDSOFT para o problema da composição de processos de software são: a Engenharia de Linha de Processos de Software (ELPS); a gestão de informações de contexto; e a combinação destas duas estratégias em um método para Linha de Processos baseada em Contexto (LPBC).

Dentro do escopo do Projeto CDSOFT, a COMPOOTIM é o foco desta tese e está voltada para a maximização da colaboração nos processos compostos para os projetos de software. A COMPOOTIM funcionou como uma instância completa do Projeto CDSOFT, pois construiu e exercitou todas as etapas da visão de pesquisa do projeto, com foco no aspecto da colaboração.

1.5. Objetivo

O objetivo geral desta tese de doutorado é *Apoiar o gerente de projeto na composição de um processo específico para o contexto do projeto, otimizando a colaboração.*

Este objetivo geral pode ser decomposto nos seguintes objetivos específicos:

- (i) Desenvolver os instrumentos para planejar, compor, otimizar e acompanhar a colaboração em processos de software;
- (ii) Definir uma sistemática de composição e otimização de processos de software que leve em consideração o aspecto da colaboração;
- (iii) Especificar e desenvolver ferramental de apoio à sistemática proposta com mecanismos de composição, otimização e visualização;
- (iv) Avaliar a viabilidade e aplicabilidade das soluções construídas.

1.6. Metodologia de Pesquisa

A metodologia de pesquisa aplicada ao desenvolvimento deste trabalho seguiu as seguintes etapas (Figura 1.1):



Figura 1.1 – Metodologia de Pesquisa

(i) *Escolha do tema e revisão da literatura*

Neste trabalho inicialmente definiu-se como tema a reutilização de processos de software, considerando a diversidade de processos. Em particular, a diversidade de modelos de desenvolvimento de software chamou a atenção em um primeiro momento.

Em seguida, uma revisão convencional da literatura sobre este tema foi realizada. Esta revisão mostrou que os modelos de desenvolvimento de software orientado ao planejamento, ágil e livre tiveram, na última década, um enorme impacto e a sua perspectiva de evolução futura é igualmente promissora (EBERT, 2007, TAURION, 2004, THEUNISSEN *et al.*, 2008). Porém, nenhum deles é efetivo para todos os projetos, pois cada um dos modelos funciona melhor ou enfrenta dificuldades diante de determinadas características dos projetos de desenvolvimento. Cada um com as suas peculiaridades, seus casos de sucesso e seus desafios, os modelos de desenvolvimento seguiram caminhos distintos, mas diversos autores defendem que eles são compatíveis e podem ser conciliados (AMBLER, 2008, ARIMOTO *et al.*, 2009, BARBIERI, 2010, BARNETT, 2004, BETTIO *et al.*, 2011, BOEHM e TURNER, 2003, COHAN e GLAZER, 2009, GLASS, 2001, GLAZER, 2010, GLAZER *et al.*, 2008, HANSSON *et al.*, 2006, HIGHSMITH, 2001, KAHKONEN e ABRAHAMSSON, 2004, MARCAL e FURTADO, 2010, MARCAL *et al.*, 2010, MELLOR *et al.*, 2002, ORR, 2002, PATEL *et al.*, 2006, PAULK, 2001, REIFER, 2002, 2003, RONG *et al.*, 2010, SALGADO *et al.*, 2010, SANTANA *et al.*, 2009, TURK *et al.*, 2002, TURNER e JAIN, 2002, VINEKAR *et al.*, 2006, WARSTA e ABRAHAMSSON, 2003).

A conciliação é definida como “o ato de restabelecer relações normais entre os beligerantes” ou a “harmonização de litigantes” (FERREIRA, 2009). No

desenvolvimento de software, a conciliação é uma tentativa de aproximar os diferentes modelos de desenvolvimento para a definição de processos de software mais eficazes. Assim, a pesquisa sobre conciliação tenta identificar sob que condições cada um dos modelos funciona melhor e compreender como o melhor de cada um pode ser combinado (JIANG e EBERLEIN, 2008).

Com base no tema de pesquisa escolhido e na revisão da literatura realizada, o objetivo desta tese foi definido. Vale destacar que a revisão da literatura continuou até a conclusão deste trabalho.

(ii) Quasi-Revisão sistemática

Com o objetivo de estudar as formas de acomodar as particularidades de cada modelo, uma *quasi*-revisão sistemática da literatura³ (BIOLCHINI *et al.*, 2005, KITCHENHAM, 2004, PAI *et al.*, 2004) sobre a conciliação dos modelos de desenvolvimento de software foi planejada e conduzida.

O objetivo da *quasi*-revisão sistemática da literatura realizada foi caracterizar a conciliação dos modelos de desenvolvimento de software orientado ao planejamento, ágil e livre. A *quasi*-revisão sistemática foi realizada em abril/2009, envolvendo quatro bases de dados digitais. Os seus resultados foram publicados em (MAGDALENO *et al.*, 2009b, 2012c).

Os resultados obtidos com esta *quasi*-revisão sistemática mostram que: a comunidade tem sugerido diferentes abordagens que conciliem os modelos de desenvolvimento; existem distintos níveis - grupo, organização e processo - de conciliação; poucos estudos lidam com a conciliação entre os três modelos de desenvolvimento; uma quantidade significativa de trabalhos trata a conciliação entre o modelo de desenvolvimento orientado ao planejamento e o ágil; grandes organizações (tais como Microsoft, Motorola e Philips) estão interessadas em combinar estes modelos; e a conciliação entre os modelos de desenvolvimento de software ainda é uma questão em aberto, uma vez que é uma área emergente de pesquisa e a maioria das propostas está em um estágio inicial.

³ Kitchenham (2004) define uma revisão sistemática da literatura como um “meio de identificar, avaliar e interpretar toda pesquisa disponível relevante a uma questão, ou área, ou fenômeno de interesse de uma pesquisa particular”. Apesar de preservar o mesmo formalismo e seguir este mesmo processo da revisão sistemática, a presente revisão caracteriza-se como uma *quasi*-revisão sistemática (TRAVASSOS *et al.*, 2008), pois o objetivo deste estudo é caracterizar a área e, portanto, não envolve a comparação de alternativas de solução.

Estrapolando estes resultados, foi possível concluir que a solução para a integração dos universos representados por estes modelos de desenvolvimento é o balanceamento das suas características. Em particular, a colaboração foi o aspecto que se destacou, pois ela está presente nos três modelos, ainda que com variações de ênfase e forma. Assim, a questão da diversidade da colaboração nos processos de software foi formulada como a questão de pesquisa desta tese.

(iii) Formatação do Projeto CDSOFT e definição do foco da tese

A partir do objetivo definido, com o corpo de conhecimento estabelecido e com os resultados obtidos da *quasi*-revisão sistemática, surgiu o arcabouço geral do Projeto CDSOFT. Enquanto o Projeto CDSOFT aborda o problema da composição de processos de software de forma abrangente, o foco desta tese foi direcionado especificamente para instanciar a visão do Projeto CDSOFT para o aspecto da colaboração.

Assim, foi definido o foco da tese na colaboração e o desenvolvimento das soluções para o planejamento, composição, otimização e acompanhamento da colaboração foram trabalhados no contexto do Projeto CDSOFT.

(iv) Elaboração da proposta de tese

A proposta de tese inicial foi defendida no exame de qualificação (MAGDALENO, 2010a) e discutida com a comunidade científica de Engenharia de Software no *Doctoral Symposium of International Conference on Software Engineering* (ICSE).

Esta discussão contribuiu para revisão e detalhamento da proposta inicial, que apresentava apenas a visão geral do trabalho sem especificar o funcionamento da COMPOOTIM, ao mesmo tempo em que demonstrou a relevância do tema.

(v) Detalhamento da solução

Nesta etapa, os instrumentos da solução COMPOOTIM foram especificados, construídos e aplicados.

Em particular, as ferramentas de apoio (EvoITrack-SocialNetwork e COMPOOTIM) foram desenvolvidas ao longo de boa parte da tese, em conjunto com os demais componentes da proposta.

Para exercitar a definição da solução também foi criado um exemplo que incluiu a definição de um conjunto de características e componentes de processos, das dimensões, informações, situações e regras de contexto para o domínio de processos de software. Este exemplo foi todo modelado no ferramental de apoio desenvolvido (MAGDALENO *et al.*, 2011b).

(vi) Avaliação da solução

Para avaliar os principais componentes da solução proposta, dois estudos exploratórios no contexto acadêmico e um estudo de caso no contexto real da indústria foram planejados e executados. Os resultados dos estudos realizados representam um indício importante da viabilidade e aplicabilidade da COMPOOTIM.

1.7. Organização do Trabalho

Esta tese está organizada em 7 capítulos. Este capítulo introdutório apresentou as principais ideias que motivaram o desenvolvimento desta tese de doutorado, o problema de pesquisa que se buscou resolver, o enfoque de solução e os objetivos da tese. Estes tópicos serão refinados ao longo dos próximos capítulos.

O **Capítulo 2** aborda a importância, os benefícios e os desafios da **colaboração**. Em particular, trata-se a colaboração no contexto do desenvolvimento de software. Em seguida, são estudados os temas que podem contribuir para planejar e acompanhar a colaboração, tais como os modelos de maturidade e as redes sociais.

No **Capítulo 3** complementa o referencial teórico desta tese focando na **reutilização de processos de software**. Dentre as estratégias para reutilização de processos – adaptação do processo padrão e composição de processos – a composição é adotada neste trabalho e detalhada neste capítulo. Assim, as técnicas para composição de processos – componentes e linha de processos - são apresentadas.

No **Capítulo 4** é descrito o contexto de desenvolvimento deste trabalho, que é o **Projeto CDSOft**. São apresentadas a sua visão geral e escopo, bem como introduzidas diferentes estratégias de solução aplicadas pelo projeto. O escopo desta tese é então delimitado no contexto do Projeto CDSOft e em

relação aos demais trabalhos de pesquisa em desenvolvimento pelo projeto.

O **Capítulo 5** detalha a proposta da **COMPOOTIM** para o planejamento, composição, otimização e acompanhamento da colaboração em processos de software. Em particular, são apresentados os instrumentos propostos ou utilizados para apoiar cada uma destas etapas.

No **Capítulo 6**, é apresentado o planejamento e analisados os resultados da execução de dois **estudos** exploratórios acadêmicos e um estudo de caso em um contexto real da indústria para avaliar as soluções da COMPOOTIM.

Finalmente, o **Capítulo 7** apresenta as **conclusões** e contribuições desta tese, relatando as limitações identificadas e indicando a continuação da pesquisa, enumerando possíveis trabalhos futuros.

Os **Apêndices I a IV** descrevem os **instrumentos utilizados no estudo** exploratório realizado para avaliar a ferramenta EvoTrack-SocialNetwork.

2. Colaboração

Neste capítulo é discutida a importância da colaboração, apresentando os seus benefícios e desafios. Em particular, aborda-se a colaboração no contexto do desenvolvimento de software. Em seguida, são consideradas formas de estimular e explicitar a colaboração, tais como os modelos de maturidade em colaboração e as redes sociais.

As organizações vêm recorrendo à colaboração entre seus profissionais para fins de qualidade, produtividade e compartilhamento de conhecimento (ALONSO *et al.*, 2010). Apesar de reconhecerem que a colaboração é vantajosa, ainda não sabem como incentivá-la (BORRELLI *et al.*, 1995), imaginando que ela se manifestará naturalmente. Na execução da colaboração de forma *ad-hoc* ou espontânea, a qualidade da colaboração é difícil de ser determinada e, geralmente, não existem objetivos ou produtos claramente definidos ou resultados esperados de sua execução para que se possa avaliá-la (DEN HENGST *et al.*, 2006).

Em particular, o desenvolvimento de software é um processo complexo que envolve a colaboração de diversas pessoas, com múltiplos talentos, durante um período de tempo, para atingir um objetivo comum (BARTHELMESS, 2003, CUGOLA e GHEZZI, 1998). Assim, o desenvolvimento de software é considerado um típico exemplo de trabalho colaborativo (ALTMANN e POMBERGER, 1999, DEMARCO e LISTER, 1999, MISTRİK *et al.*, 2010, OMORONYIA *et al.*, 2010, SOUZA *et al.*, 2011, YILDRIM, 2006). Whitehead *et al.* (2010) chegam a afirmar que “a colaboração é pervasiva por toda a Engenharia de Software”, pois a maioria dos projetos de software ocorre dentro de um processo intensivamente colaborativo. Portanto, compreender como as pessoas se organizam e interagem umas com as outras enquanto desenvolvem software é importante para melhorar a produtividade e qualidade (VALETTO *et al.*, 2007).

Apesar da sua importância, a colaboração ainda é um desafio em diversas áreas, incluindo a Engenharia de Software (ES), em termos de quais práticas, processos e ferramentas são capazes de estimulá-la e acompanhar sua execução (MISTRİK *et al.*, 2010). Tal fato é reforçado pelas pesquisas sobre desenvolvimento distribuído de software (CATALDO e HERBSLEB,

2008, HERBSLEB *et al.*, 2005, OMORONYIA *et al.*, 2010), métodos ágeis (COCKBURN e HIGHSMITH, 2001, COCKBURN, 2001), software livre (RAYMOND, 2001) e ecossistemas (BOSCH e BOSCH-SIJTSEMA, 2010), onde o foco nas pessoas (nos seus talentos, habilidades e conhecimentos) e a preocupação com a colaboração e comunicação aparecem de forma recorrente.

Para estimular a colaboração, Araujo e Borges (2007) defendem que é preciso aumentar a sua visibilidade de forma que os membros da organização atinjam maior compreensão sobre como se comportam as interações entre eles e se motivem a participar das mesmas. Este aumento de visibilidade pode ser alcançado através da oferta de mecanismos que permitam que os atores tenham informação sobre como a colaboração ocorre em um processo de software.

Sendo a colaboração um aspecto fundamental na execução de processos de software, deve ser planejada e acompanhada. Planejar a colaboração significa permitir às organizações de software estabelecer objetivos e definir ações para atingir níveis desejados de colaboração em um processo. O acompanhamento diz respeito a avaliar se o nível colaboração planejado é atingido, identificando oportunidades para a tomada de ações corretivas. Neste capítulo, modelos de maturidade de colaboração e técnicas de visualização e análise de redes sociais de execução de processos são apresentados como abordagens em potencial para planejar e acompanhar a colaboração em processos de software.

O restante deste capítulo está organizado da seguinte forma: a Seção 2.1 define colaboração e aborda os seus benefícios e desafios; a Seção 2.2 é dedicada aos aspectos que caracterizam a colaboração (comunicação, coordenação, memória e percepção); na Seção 2.3 são discutidos brevemente os principais modelos de maturidade em colaboração existentes na literatura; a Seção 2.4 é dedicada às redes sociais como instrumentos para o entendimento da colaboração existente entre os membros de um grupo; por fim, a Seção 2.5 tece considerações finais ao capítulo.

2.1. Definição

A palavra colaboração deriva do latim *com* e *laborare* e significa trabalho em conjunto. Existe uma variedade de definições e entendimento sobre o significado da colaboração e falta um consenso sobre o assunto (ALONSO *et al.*, 2010, THOMSON *et al.*, 2009). Uma definição simples, objetiva e comumente aceita é oferecida por Vreede e Briggs (2005): “articulação de um esforço comum em direção a um objetivo”. Neste trabalho, adota-se uma definição similar para a colaboração: “trabalho em conjunto de duas ou mais pessoas para a realização de objetivos comuns” (Adaptado de FERREIRA, 2009, MINICUCCI, 2001).

As razões pelas quais um determinado grupo de pessoas se reúne para realizar uma determinada tarefa podem ser as mais variadas possíveis. Quando uma tarefa ou problema é grande, complexo ou requer múltiplas competências, é necessário unir os esforços de várias pessoas. Em geral, grupos se formam para a construção de um produto que pode ser tão concreto como um texto, um software ou o projeto de um artefato; tão abstrato como uma decisão ou a formação de um conhecimento comum – um aprendizado – sobre um determinado assunto (ARAUJO e BORGES, 2007).

Durante o tempo no qual as pessoas estão trabalhando em conjunto, elas estão organizadas em grupos. Há uma expectativa de que os grupos tendam a superar o desempenho quantitativo e qualitativo de indivíduos agindo sozinhos. Os grupos podem alcançar uma compreensão que nenhum dos seus membros possuía previamente e que não poderia ter sido obtida caso tivessem trabalhado de forma isolada, pois representa mais do que a soma individual das partes (ALEIXO, 2003, FUKS *et al.*, 2003b, HAINES e VEHRING, 2012, DEN HENGST *et al.*, 2006, MINICUCCI, 2001, NUNAMAKER *et al.*, 2001, SARMENTO, 2002, VIVACQUA e GARCIA, 2011).

De modo geral, as principais vantagens da colaboração são: redução no tempo necessário para a execução de tarefas; melhoria da capacidade de resolver problemas complexos; aumento da capacidade criativa para gerar alternativas; discussão das vantagens e desvantagens de cada alternativa para selecionar as viáveis e tomar decisões; melhoria na comunicação;

aprendizagem; satisfação pessoal; e estímulo à inovação (ALEIXO, 2003, ALONSO *et al.*, 2010, DEMARCO e LISTER, 1999, HARDINGHAM, 2000).

Apesar destes potenciais benefícios, alcançar uma colaboração efetiva ainda é um desafio para muitas organizações (ALONSO *et al.*, 2010, BORRELLI *et al.*, 1995, VREEDE e BRIGGS, 2005). Por isso, é importante determinar quando a colaboração é realmente necessária (DEAN *et al.*, 2006) para usá-la de forma eficiente (DEN HENGST *et al.*, 2006). A colaboração só deve ser adotada quando tem potencial para produzir melhores resultados do que os indivíduos trabalhando isoladamente. Este potencial da colaboração depende de alguns critérios, tais como alinhamento de objetivos, resultados esperados claramente definidos, patrocínio e disponibilidade de orçamento (HANSEN, 2009).

Em um contexto desfavorável, a colaboração também poderá enfrentar desafios (SARMENTO, 2002, VIVACQUA e GARCIA, 2011): lentidão na execução das tarefas de forma colaborativa; falta de coordenação do trabalho; custo elevado da tomada de decisão; e resistência ao compartilhamento de conhecimento. Nestes casos, os recursos são consumidos em atividades ineficientes, sem atingir os benefícios desejados com a colaboração.

2.2. Aspectos de Apoio à Colaboração

A área de pesquisa CSCW (*Computer Supported Cooperative Work*) (ALTMANN e POMBERGER, 1999, ELLIS *et al.*, 1991, GEROSA *et al.*, 2006, GREIF, 1988, KHOSHAFIAN e BUCKIEWICZ, 1995, MISTRİK *et al.*, 2010, PIMENTEL e FUKS, 2011) possui uma extensa literatura que estuda como oferecer apoio computacional à colaboração. Nesta área de pesquisa, o corpo de conhecimento sobre apoio à colaboração pode ser estruturado sob quatro aspectos - comunicação, coordenação, memória e percepção (ARAUJO e BORGES, 2007, ARAUJO, 2000) (Figura 2.1) - que tradicionalmente são usados nas ferramentas de *groupware*⁴. Alternativamente, no modelo de colaboração 3C (FUKS *et al.*, 2003b, 2011), estes aspectos são resumidos em três - comunicação, coordenação e cooperação – onde este último item

⁴ *Groupware* pode ser definido como “qualquer tecnologia computacional que auxilie grupos a trabalharem cooperativamente através de mídia digital” (KHOSHAFIAN e BUCKIEWICZ, 1995).

concentra os aspectos de memória de grupo e percepção.

Os aspectos de apoio não podem ser considerados isoladamente, pois se encontram intimamente dependentes e relacionados entre si, conforme o esquema apresentado na Figura 2.1. Para colaborar, os indivíduos têm que trocar informações (*comunicação*), organizar-se (*coordenação*) e operar em conjunto em um espaço de trabalho coletivo (*memória*). Através da *percepção*, o indivíduo se informa sobre o que está acontecendo e adquire as informações necessárias para o seu trabalho (ARAUJO e BORGES, 2007, ARAUJO, 2000, DIAS, 1998, FUKS *et al.*, 2003b).

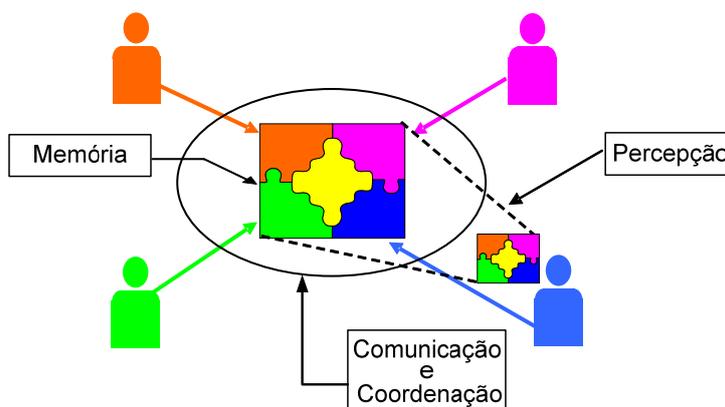


Figura 2.1 – Esquema geral dos aspectos de apoio à colaboração (ARAUJO e BORGES, 2007)

A comunicação, coordenação, memória e percepção são os aspectos que ajudam a caracterizar as interações e a colaboração em um grupo. Portanto, estes aspectos são usados neste trabalho para planejar e acompanhar a colaboração.

2.2.1. Comunicação

O primeiro obstáculo à colaboração é vencer a distância entre os membros do grupo, ou seja, estabelecer a comunicação entre as partes envolvidas para que elas possam obter sucesso ao trabalhar em grupo (ARAUJO e BORGES, 2007, MEIER *et al.*, 2007). A comunicação compreende o intercâmbio de informações entre pelo menos dois indivíduos, com a compreensão da mesma pelos envolvidos. A comunicação apoia a criação de um entendimento comum e compartilhado (OLIVEIRA, 2009). Os membros de um grupo precisam se comunicar regularmente para organizar o trabalho, negociar, designar tarefas, tomar decisões, firmar compromissos e resolver problemas.

Dentro de um grupo, a comunicação pode ser realizada em tempo real (síncrona) ou em momentos diferentes (assíncrona). Na forma síncrona, os

interlocutores estão presentes simultaneamente e disponíveis e a mensagem enviada é recebida imediatamente. Na forma assíncrona, o tempo é mais flexível: a mensagem pode ser recebida em um momento posterior indeterminado (ELLIS *et al.*, 1991, FUKS *et al.*, 2003a).

A comunicação assíncrona normalmente é utilizada quando se deseja valorizar a reflexão dos participantes, pois estes terão mais tempo antes de agir. Já na comunicação síncrona, valoriza-se a velocidade da interação, visto que o tempo de resposta entre a ação de um participante e a reação de seus companheiros é curto (FUKS *et al.*, 2003a).

2.2.2. Coordenação

O trabalho em grupo demanda um esforço adicional para a coordenação de seus membros e suas atividades. Ou seja, é necessário um trabalho de articulação para que a colaboração possa ser obtida a partir da soma dos trabalhos individuais. A coordenação entre as atividades do grupo é necessária para que o trabalho de um não afete indevidamente o trabalho dos outros, visto que existem interdependências entre essas atividades (HAINES e VEHRING, 2012, KRAUT e STREETER, 1995, MEIER *et al.*, 2007).

A coordenação organiza o grupo para evitar que esforços de comunicação sejam perdidos e para que as tarefas sejam realizadas na ordem correta, no tempo previsto e cumprindo as restrições e objetivos, a fim de evitar que os participantes se envolvam em tarefas conflitantes ou repetitivas (FUKS *et al.*, 2003b, GEROSA *et al.*, 2003, VIVACQUA e GARCIA, 2011). A coordenação visa manter a “vida do grupo”, através de estímulos às contribuições de cada participante, agendamento de eventos e estabelecimento de um ritmo aos trabalhos e encontros (SANTORO, 2001).

A coordenação envolve a pré-articulação, o acompanhamento e a pós-articulação das tarefas. A pré-articulação engloba as ações necessárias para planejar a colaboração: identificação dos objetivos, mapeamento destes objetivos em tarefas, seleção dos participantes e a distribuição das tarefas entre eles. Neste planejamento, leva-se em consideração a natureza da tarefa em questão, os recursos individuais e as áreas de conhecimento de cada participante. Para cada tarefa, é preciso definir: o que deve ser feito, quem deve executá-la e quando deve ser feita (HAINES e VEHRING, 2012).

O acompanhamento significa controlar a execução das atividades durante o processo, para garantir a produtividade e o sucesso dos objetivos do grupo. A pós-articulação envolve a avaliação das tarefas realizadas e a documentação da colaboração (ARAUJO e BORGES, 2007, DIAS, 1998, FUKS *et al.*, 2003b).

2.2.3. Memória de grupo

Comunicação e coordenação, apesar de vitais, não são suficientes para a colaboração. Os grupos também precisam produzir, manipular, organizar e compartilhar diferentes tipos de informação relacionada às atividades sendo realizadas (ELLIS *et al.*, 1991). A memória de grupo é o armazenamento dos dados relativos ao desenvolvimento da atividade colaborativa.

Além de construir e refinar artefatos, os participantes também compartilham reflexões sobre o trabalho sendo produzido. Assim, é importante preservar não só o conhecimento formal (embutido no histórico e organização dos artefatos) como também o conhecimento informal sobre a criação dos produtos, as ideias, fatos, questões, pontos de vista, conversas, discussões e decisões que aconteceram no decorrer do trabalho. Ainda que este conhecimento informal seja mais difícil de ser capturado, é ele que permite recuperar o histórico da discussão e o contexto em que as decisões foram tomadas (ARAUJO e BORGES, 2007, DIAS, 1998, FUKS *et al.*, 2003b, NONAKA e TAKEUCHI, 1995).

2.2.4. Percepção

O fenômeno de criação em grupo se dá progressivamente através da geração de novas ideias baseadas nas contribuições já elaboradas pelo grupo. Perceber e entender as atividades realizadas por outros participantes são requisitos fundamentais para que haja comunicação e interação em um grupo em direção a um objetivo comum (DAVID, 2004). A percepção é o “entendimento das atividades dos outros para fornecer contexto a sua própria atividade” (DOURISH e BELLOTTI, 1992). A percepção denota as práticas pelas quais os atores, de forma tácita e não intrusiva, alinham e integram as suas atividades (SCHMIDT, 2002).

Manter a percepção é considerado um fator crítico para garantir que os membros da equipe sejam capazes de coordenar os seus esforços (DOURISH

e BELLOTTI, 1992, HAINES e VEHRING, 2012). A percepção compreende as informações que os membros do grupo mantêm uns a respeito dos outros, sobre as atividades e os artefatos do esforço colaborativo compartilhado (SCHMIDT, 2002). Por exemplo, através da percepção, os indivíduos podem tomar ciência do papel de cada um dentro do grupo, do que fazer, como proceder, qual o resultado das suas ações, até onde atuar, presença ou status dos membros do grupo, possíveis conflitos entre os membros do grupo, quais as modificações realizadas em um artefato, como estão os recursos compartilhados pelo grupo e etc (GUTWIN e GREENBERG, 1999, OMORONYIA *et al.*, 2010, STEINMACHER *et al.*, 2012).

Cada membro deve conhecer o progresso do trabalho dos companheiros e dar visibilidade das suas próprias ações: o que foi feito, como foi feito, o que falta para o término, quais são os resultados preliminares e etc. A percepção das contribuições e dos resultados gerados pelas atividades alheias ajuda os membros do grupo a oferecer contribuições com maior segurança quanto à necessidade e relevância para o produto comum (ARAUJO e BORGES, 2007, DOURISH e BELLOTTI, 1992, GEROSA *et al.*, 2003). Quando os membros não têm conhecimento sobre o que está sendo desenvolvido pelos outros, o trabalho resultante pode não apresentar coesão e não representar as ideias do grupo como um todo (SANTORO, 2001).

2.3. Modelos de Maturidade em Colaboração

Modelos de maturidade foram definidos para avaliar a maturidade em um determinado domínio com base em um conjunto de critérios (DE BRUIN *et al.*, 2005). Um modelo de maturidade consiste em um *framework* que descreve, para uma área de interesse específica, uma sequência progressiva de níveis de maturidade em que as atividades nesta área podem ser realizadas (ALONSO *et al.*, 2010, BECKER *et al.*, 2009, PÖPPELBUß e RÖGLINGER, 2011).

Essencialmente, um modelo de maturidade pode ser usado: (i) para avaliar e compreender a situação atual da organização, identificando oportunidades para otimização; (ii) para estabelecer objetivos e recomendar ações de melhoria para aumentar a capacidade de uma área específica da

organização; (iii) para orientar a definição de processos; (iv) como um instrumento de planejamento e controle para acompanhar o sucesso das ações tomadas (BECKER *et al.*, 2009, HAIN e BACK, 2011, HAIN, 2010).

Em diferentes domínios, como ES (CHRISSIS *et al.*, 2006, PAULK *et al.*, 1994), Gestão de Conhecimento (EHMS e LANGEN, 2002, TATA, 2010), Gestão de TI (BECKER *et al.*, 2009) e BPM (*Business Process Management*) (FISHER, 2004, ROSEMANN *et al.*, 2004), tanto na academia quanto na indústria, modelos de maturidade têm sido propostos como uma forma de organizar um corpo de conhecimento e avaliar as organizações em sua capacidade ou maturidade de execução das práticas propostas no modelo. A base para a maioria destes modelos foi o CMM (*Capability Maturity Model*) (PAULK *et al.*, 1994) - originalmente criado pelo SEI (*Software Engineering Institute*) em 1991 e posteriormente evoluído para o CMMI (*Capability Maturity Model Integration*) (CHRISSIS *et al.*, 2006), desenvolvidos para avaliar a maturidade ou capacidade dos processos de software das organizações. Alguns destes domínios e seus respectivos modelos de maturidade são descritos em (MAGDALENO *et al.*, 2011a, 2011c).

Sob o ponto de vista da colaboração, a maturidade é entendida como a “máxima capacidade atual de uma equipe para colaborar, onde os membros da equipe se comunicam efetivamente, chegam a um entendimento compartilhado e ajustam as suas tarefas e comportamentos para produzir resultados de alta qualidade” (BOUGHZALA e DE VREEDE, 2012).

Como a mudança para um ambiente colaborativo não é trivial, ela se concretiza de forma gradual, pois ao longo do tempo as organizações vão adquirindo maturidade no uso efetivo da colaboração. Assim, acredita-se que um modelo de maturidade seja um instrumento adequado para planejar e explicitar como a colaboração deve acontecer dentro de uma organização.

Em um trabalho sobre o estado da arte dos modelos de maturidade em colaboração, Hain e Back (2009) fizeram uma revisão rigorosa da literatura e identificaram 55 modelos de maturidade nas áreas de colaboração, gestão de conhecimento e *e-learning*. Entre estes modelos, existem tanto propostas acadêmicas quanto práticas e muitos destes modelos são derivados do CMM. Porém, a maioria dos modelos é voltada para gestão de conhecimento e *e-learning* e observa-se que existe uma escassez de modelos de maturidade

especificamente voltados à colaboração. Os modelos identificados são brevemente apresentados e discutidos a seguir.

- **ECMM** (*Enterprise Collaboration Maturity Model*) (ALONSO *et al.*, 2010) - visa avaliar a colaboração e interoperabilidade em organizações que operam em redes virtuais ou ecossistemas. O modelo é composto por quatro níveis de maturidade e 24 áreas de processo.
- **SiMMCo** (*Situational Maturity Model for Collaboration*) (HAIN e BACK, 2011, HAIN, 2010) - tem como propósito ajudar os gestores a analisar a situação atual da organização, identificar se a organização tem capacidade para trabalhar de forma colaborativa ou o que deve ser implementado para atingir essa capacidade. Este modelo ainda não foi efetivamente criado e os autores apresentam apenas a ideia do modelo, mas a proposta é que ele seja construído com base nos fatores críticos de sucesso extraídos da literatura e nos estudos de caso que serão conduzidos.
- **CEMM** (*Collaboration Engineering Maturity Model*) (SANTANEN *et al.*, 2006) - modelo para implantar a Engenharia de Colaboração⁵ dentro de uma organização. O modelo é complexo e bidimensional: a primeira dimensão (eixo x) representa as fases da abordagem; e a segunda dimensão (eixo y) indica os níveis de maturidade correspondentes a cada fase (SANTANEN *et al.*, 2006). As fases da abordagem devem ser seguidas em sequência para que seja possível projetar e implantar um processo de colaboração que permita à organização sustentar e apoiar as práticas de colaboração. Estas fases incluem: Entrevista de campo, *Design*, Transição, Implementação e, finalmente, Uso organizacional sustentado.
- **Col-MM** (*Collaboration Maturity Model*) (BOUGHZALA e DE VREEDE, 2012) proposto mais recentemente, avalia a qualidade da colaboração de um grupo dentro da organização. O Col-MM foi desenvolvido durante uma série de reuniões com profissionais especialistas em colaboração. O Col-MM inclui um modelo, um método de análise e um instrumento de avaliação (questionário). O modelo é composto por quatro níveis de maturidade, mas estes níveis são formados por critérios e não por práticas de colaboração.

⁵ A Engenharia de Colaboração (*Collaboration Engineering – CE*) é uma nova abordagem para suporte à colaboração, assim como existiram no passado os GSSs (*Group Support Systems*). A CE está voltada para o *design* e implantação de padrões de colaboração (VREEDE e BRIGGS, 2005).

O Col-MM explora a colaboração sobre diferentes perspectivas (características de colaboração, gestão da colaboração, processo da colaboração e integração de informação e conhecimento). Este modelo ainda está em sua versão inicial, e em sua primeira aplicação em um estudo exploratório.

- **CollabMM** (*Collaboration Maturity Model*) – Proposto em um trabalho prévio (MAGDALENO, 2006) para apoiar as organizações em incorporar explicitamente aspectos de colaboração em seus processos, permitindo que a colaboração seja aplicada de forma sistemática durante a modelagem do processo (MAGDALENO *et al.*, 2009a). Desta forma, a organização pode avaliar o seu nível de maturidade em colaboração, identificar as melhores práticas para iniciar a implementação da colaboração e planejar as suas ações de melhoria futuras.

2.4. Redes Sociais

No contexto desta pesquisa, o mecanismo utilizado para explicitar e acompanhar a colaboração existente entre os participantes em projetos de desenvolvimento de software são as redes sociais. De forma simplificada, uma **rede** é “um conjunto de pontos interligados” (WWF, 2003). Formalizando esta definição, pode-se dizer que uma rede é um “conjunto de vértices ou nós com conexões entre eles, denominadas arestas” (NEWMAN, 2003). Para ilustrar estas definições, dois exemplos de redes, com representações distintas, são apresentados na Figura 2.2.

As redes ajudam na compreensão de determinados fenômenos e suas conexões. Por exemplo, mapas detalhados da Internet ajudaram a perceber a vulnerabilidade da rede aos ataques dos *hackers*. Os mapas das empresas conectadas por relações comerciais traçaram o caminho do poder e do dinheiro em Silicon Valley. Os mapas de interações entre as espécies do ecossistema mostraram o impacto destrutivo do ser humano no meio ambiente. Até mesmo os mapas dos genes trabalhando em conjunto dentro de uma célula ajudaram no entendimento sobre o funcionamento do câncer (BARABASI, 2003). Outras importantes aplicações de redes, tais como *marketing*, vendas e etc. também podem ser mencionadas (MONCLAR *et al.*, 2011).

Em um diagrama de rede, como os exemplos apresentados na Figura 2.2, há necessariamente pontos e linhas. Os pontos representam as unidades que compõem a rede: pessoas, organizações, equipamentos, locais e etc. e são denominados de nós ou vértices. As linhas representam as relações entre esses elementos e podem ser canais de comunicação, estradas, dutos, fios e etc. Estas linhas recebem o nome de relacionamentos, arestas, ligações ou conexões (WWF, 2003).

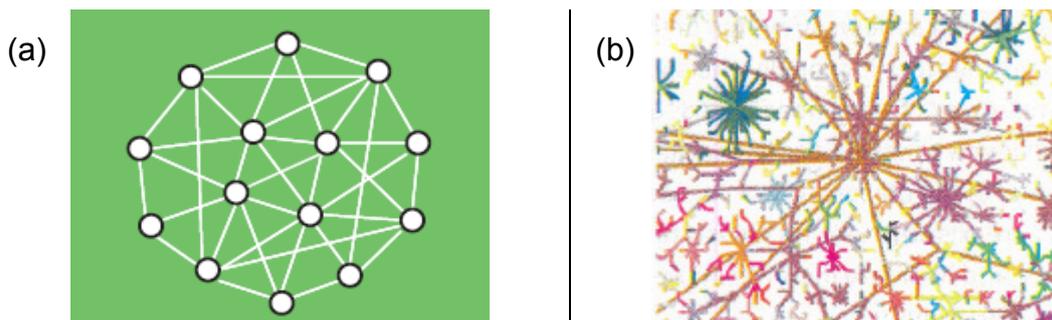


Figura 2.2 – Exemplos de redes (BARABASI e BONABEAU, 2003, WWF, 2003)

Em particular, as **redes sociais** consistem em um “conjunto finito de atores e as relações definidas entre eles” (WASSERMAN e FAUST, 1994). Em uma rede social, os nós representam os atores e as arestas correspondem aos possíveis relacionamentos entre eles. A semântica do relacionamento depende da análise que se deseja conduzir nesta rede.

Alguns exemplos de redes sociais, com semânticas distintas, são (ALBERT e BARABASI, 2002, BARABASI e BONABEAU, 2003): rede de relações de amizade; casamentos entre famílias; comunidades de negócio; rede de contatos sexuais; rede de colaboração entre atores que contracenaram juntos em filmes; rede de colaboração científica representada através da coautoria em artigos acadêmicos; e a rede de comunicação (ligações telefônicas, e-mail ou *instant messaging*).

As redes sociais são dinâmicas, sofrem mudanças ao longo do tempo, e sua evolução é baseada na entrada e saída de nós na rede. Quando novos nós são adicionados na rede, eles decidem com quem se conectar e existe uma preferência pelos nós mais visíveis, que já têm mais conexões. Ou seja, quanto mais conectado é um nó, maior probabilidade ele tem, pela extensão de sua rede, de fazer novas conexões. Desta forma, um nó mais popular ou que se destaca na rede devido a uma grande quantidade de conexões – um *hub* – passa a ter cada vez mais relacionamentos ao longo do tempo (BARABASI,

2003). Entretanto, a rede não é permanente. Um nó simples pode transformar-se em *hub*, enquanto outros *hubs* morrem. Os *hubs* são importantes na rede, pois criam caminhos mais curtos entre os nós da rede, sendo capazes de proliferar rapidamente ideias, informações, doenças e etc.

2.4.1. Propriedades Topológicas

As redes sociais são baseadas na construção teórica da sociologia (DEGENNE e FORSE, 1999) e na fundamentação matemática da teoria de grafos (SZWARCFITER, 1986). Grafos ou redes têm propriedades topológicas que explicam a sua estrutura (WASSERMAN e FAUST, 1994). O entendimento destas propriedades topológicas e das leis e fenômenos que as caracterizam é fundamental para a tarefa de análise das redes sociais.

Formalmente, uma rede é representada através de um grafo G composto por dois conjuntos: um conjunto (N) de objetos (nós) e um conjunto (E) de pares relacionados (arestas). Assim, dado o grafo G apresentado na Figura 2.3, temos que $N = \{a, b, c, d, e, f\}$; e $E = \{(a,c), (b,c), (b,d), (b,f), (c,d), (d,e), (e,f)\}$. A partir daí, podem ser observadas as seguintes propriedades:

- **Nós adjacentes:** dois nós são adjacentes ou vizinhos se existe uma aresta entre eles. Assim, considerando o exemplo da Figura 2.3, é possível afirmar que o nó a é vizinho de c , pois existe uma aresta ligando-os. Porém, o nó d não é vizinho do nó f , pois eles não estão diretamente relacionados;
- **Grau:** o grau de um nó ($g(n)$) corresponde ao número total de arestas incidentes a ele. Nem todos os nós de uma rede têm o mesmo número de arestas. Por exemplo, na Figura 2.3, alguns nós têm um grau pequeno, enquanto outros têm um grau mais alto. Assim, $g(a) = 1$, pois o nó a tem apenas 1 aresta com c e $g(b) = 3$, pois o nó b tem arestas com outros três nós (c, d, f);

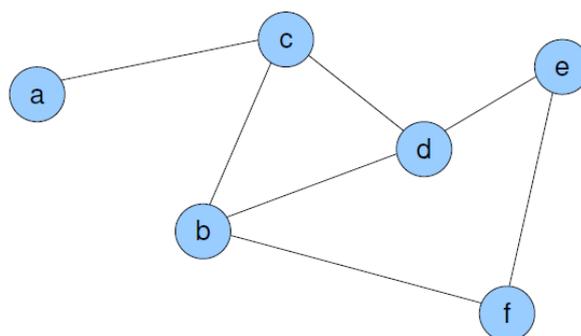


Figura 2.3 – Exemplo de rede para estudo das propriedades topológicas

- **Caminho:** um caminho é uma sequência de nós de um ponto ao outro do grafo. Pode existir mais de um caminho ligando dois nós. Então, para percorrer o grafo da Figura 2.3 de *a* até *f* é possível escolher um caminho passando por a-c-d-e-f ou passando por a-c-b-f;
- **Comprimento de caminho:** o comprimento de um caminho é calculado pelo número de arestas percorridas ao longo do caminho. Assim, no exemplo citado acima, os dois caminhos indicados têm comprimentos distintos. No primeiro caminho, são percorridas 4 arestas e, no segundo caminho, são percorridas apenas 3 arestas para se chegar ao nó destino. Logo, o comprimento do segundo caminho é menor;
- **Distância:** a distância corresponde ao comprimento do *menor caminho* entre dois nós, considerando todos os caminhos existentes. Ainda usando o mesmo exemplo anterior, a distância entre *a* e *f* é igual a 3.

2.4.2. Metodologia para estudo de redes sociais

Resumindo a abordagem proposta em (CROSS *et al.*, 2004), é possível estabelecer quatro passos para a metodologia de estudo de redes sociais:

(i) definição da rede que se deseja analisar: O primeiro passo é definir o objetivo da análise e estabelecer a semântica dos nós e arestas da rede que se deseja analisar. A análise da rede social será potencialmente mais benéfica em redes que tenham uma maior necessidade ou potencial de colaboração e aonde a colaboração seja importante para a organização e traga benefícios estratégicos ou operacionais;

(ii) coleta dos dados necessários: O próximo passo é a coleta de dados para a construção da rede social. Esta coleta pode ser feita através de questionários (papel, e-mail ou Internet) ou facilitada pela mineração de dados (AALST *et al.*, 2005) que busca os dados em repositórios de informação para que eles possam ser manipulados para visualização ou análise. Por exemplo, nos projetos de desenvolvimento de software livre, é comum o uso de *parsing* nas páginas dos projetos e nas suas listas de e-mail e fóruns de discussão;

(iii) visualização de redes sociais: No passo de visualização de redes sociais, é adotada a representação visual das informações, através de diferentes formas, para diminuir a sobrecarga cognitiva do usuário e facilitar a compreensão e exploração dos dados através de imagens (VIEGAS e

DONATH, 2004). A visualização aproveita a habilidade do ser humano no reconhecimento de padrões e estruturas em informações visuais. Portanto, torna mais fácil e natural a tarefa de compreensão das informações apresentadas. Entretanto, a visualização também possui limitações no caso de diagramas de grande dimensão, devido à sobreposição de elementos.

A visualização de redes sociais permite a observação de fatos e a extração de conhecimento a partir de grafos (FREITAS *et al.*, 2008). A visualização de uma rede social é útil para identificar pessoas chave, para a compreensão da estrutura de uma determinada rede social ou para comparar redes semelhantes. Tamanho, forma ou cor podem ser usados para codificar informações em um nó ou aresta, aumentando o poder expressivo da visualização (OEZBEK *et al.*, 2010).

Os mecanismos de visualização se diferenciam pelos recursos que oferecem e pelas perspectivas que exploram. Na perspectiva temporal, por exemplo, o interesse gira em torno do entendimento da dinamicidade da rede, ou seja, como as redes se desenvolvem ou mudam no decorrer do tempo.

A visualização da rede é uma informação importante, pois oferece percepção sobre o que está acontecendo com as interações ou colaboração do grupo. Entretanto, sozinha, a visualização não permite a total compreensão da colaboração existente entre os atores da rede e precisa ser complementada pela análise das redes sociais, que é o passo seguinte da metodologia.

(iv) análise de redes sociais: Por fim, o passo de análise de redes sociais utiliza as propriedades topológicas oriundas da teoria de grafos (vistas na Seção 2.4.1) para descrever a interação e a organização social de um grupo. O objetivo é entender padrões, geralmente implícitos, nos relacionamentos existentes entre os nós de uma rede social (MEIRA *et al.*, 2011). É considerada como um “raio-x da organização”, e pode ser usada como uma ferramenta de diagnóstico e planejamento para aumentar a colaboração e, conseqüentemente, o fluxo de informações na organização (PEREIRA e SOARES, 2007). Devido a sua relevância neste trabalho, este passo é aprofundado na próxima seção.

2.4.3. Análise de redes sociais

A Análise de Redes Sociais (ARS), do inglês *Social Network Analysis* (SNA) é

uma abordagem metodológica, originada das ciências sociais, que utiliza os conceitos da teoria de grafos para descrever, entender e explicar as estruturas sociais (BRANDES e WAGNER, 2003) e os relacionamentos entre os membros de uma rede (WASSERMAN e FAUST, 1994).

A ARS permite observar as relações que facilitam ou dificultam a colaboração e o compartilhamento de informação e conhecimento em uma organização. As observações oriundas da ARS podem indicar gargalos ou oportunidades para a melhoria do fluxo de informações para evitar problemas de comunicação e perda de informações e conhecimento que deveriam circular na rede. Após diagnosticar os possíveis problemas, é possível propor algumas sugestões de melhorias para ajudar a organização a alcançar uma colaboração e comunicação efetivas (CROSS *et al.*, 2004, MONCLAR *et al.*, 2011, PEREIRA e SOARES, 2007).

A ARS já foi explorada em diferentes contextos (MONCLAR *et al.*, 2011): sociais (EGLER, 2007); médicos (MOREL *et al.*, 2009); ambientais (WWF, 2003); organizacionais (CROSS *et al.*, 2004); de pesquisa científica (BIRD *et al.*, 2009a); melhorias de processos de software (NIELSEN e TJØRNEHØJ, 2005); na área de BPM como parte de uma estratégia de *Social BPM* (SWENSON *et al.*, 2011); na área de CSCW para derivar requisitos para sistemas colaborativos (PEREIRA e SOARES, 2007); e no contexto do desenvolvimento de software, conforme apresentado na Seção 2.4.4.

2.4.3.1. Propriedades de análise de redes sociais

As redes sociais podem ser analisadas através das suas propriedades (PEREIRA e SOARES, 2007, WASSERMAN e FAUST, 1994). Um dos principais usos da ARS é medir a importância de um nó dentro da rede. Um nó da rede é considerado proeminente se os seus relacionamentos o tornam particularmente visível aos outros nós da rede. Então, existem duas classes de proeminência do ator: centralidade e prestígio (WASSERMAN e FAUST, 1994).

Na centralidade, a direção das relações nas quais o ator está envolvido não importa, ou seja, um nó central é aquele que está intensivamente envolvido em relacionamentos com outros atores, seja como transmissor ou como receptor. Por sua vez, o conceito de prestígio do ator é mais restritivo, pois só reconhece as relações onde o ator é o receptor. Assim, o conceito de prestígio do ator não pode ser observado em um grafo não direcionado. Entre os tipos

de centralidade (FREEMAN, 1978, OLIVEIRA, 2012, WASSERMAN e FAUST, 1994), destacam-se:

- **Centralidade de grau** (*degree centrality*): está relacionada ao número de relações diretas que este nó mantém na rede. O nó mais central ou *hub* (BARABASI, 2003) é aquele que possui o maior grau. Como ele está em contato direto com muitos outros nós, ele acaba ocupando um lugar central na rede e desempenhando um papel importante. Em contrapartida, nós com grau pequeno ocupam uma posição periférica na rede. Usando o grafo apresentado na Figura 2.4 como exemplo, o nó mais central é o nó *c* que possui a maior centralidade de grau, que é igual a 3, pois possui 3 arestas. Os outros nós não atingem centralidade de grau igual a 3. Os nós *a* e *b* possuem centralidade de grau igual a 2 e o nó *d* possui a menor centralidade de grau desta rede que é igual a 1;

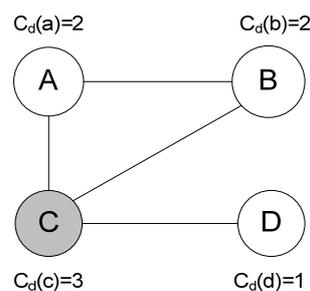


Figura 2.4 – Exemplo de centralidade de grau

- **Centralidade de proximidade** (*closeness centrality*): esta propriedade identifica uma posição estratégica na rede e é inversamente relacionada com a distância. Ela representa o quão próximo um nó se encontra em relação aos demais nós da rede. A centralidade de proximidade é calculada pelo inverso da soma das distâncias de um nó para os outros destinos possíveis. A ideia é que um nó é central se ele pode interagir rapidamente com os demais. Este nó central pode ser altamente produtivo no compartilhamento de informações com o grupo, já que ele acelera o caminho de comunicação entre os nós. Assim, quanto mais diminui a distância de um nó para o restante da rede, mais aumenta a sua centralidade de proximidade. Em contraste, quanto maior a distância entre os nós, maior a chance de perda ou má interpretação entre as informações transferidas ao longo da rede. Considerando como exemplo, a rede em formato de estrela, apresentada na Figura 2.5, o nó *b*, localizado no centro,

é adjacente a todos os outros. Logo, ele tem uma centralidade de proximidade máxima, já que partindo de b todos os outros nós podem ser alcançados em apenas 1 passo;

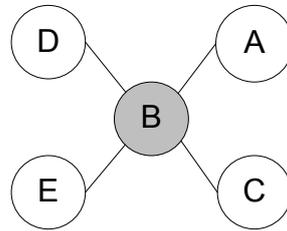


Figura 2.5 – Exemplo de centralidade de proximidade

- **Centralidade de intermediação** (*betweenness centrality*): as interações entre dois nós não adjacentes dependem dos nós que se localizam no caminho entre eles. Estes outros nós potencialmente têm controle sobre as interações entre os dois nós não adjacentes. Assim, a centralidade de intermediação é medida pelo número de vezes que o nó aparece no menor caminho entre outros nós. Esta medida quantifica o grau em que um indivíduo da rede controla ou intermedia o fluxo de informação. Por exemplo, no grafo apresentado na Figura 2.6, o nó b não é o que tem a maior centralidade de grau, mas ele exerce um papel importante nessa rede e possui uma alta centralidade de intermediação, pois é um ponto de articulação entre os nós $\{a, c, d, e, f, x\}$ e os nós $\{g, h\}$;

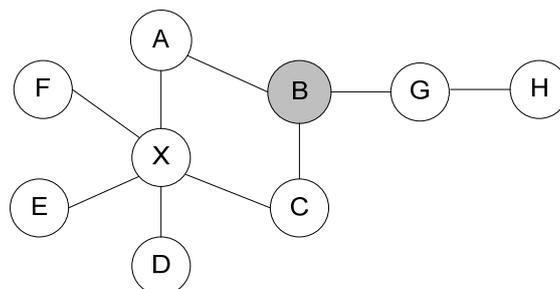


Figura 2.6 – Exemplo de centralidade de intermediação

Apesar de essas propriedades de centralidade terem foco nos nós, elas também podem ser analisadas do ponto de vista da rede como um todo ou de determinados subgrupos. Desta forma, podem ser úteis para comparar diferentes redes, desde que sejam normalizadas de acordo com o tamanho da rede. Por outro lado, outras propriedades, como a densidade da rede, exploram diretamente as propriedades gerais da rede.

- **Densidade da rede:** a densidade da rede está relacionada à quantidade de

arestas que mantêm interligado um conjunto de nós. Quanto mais arestas existir numa rede, mais densa ela será (WWF, 2003). A densidade indica a proporção entre o número de relacionamentos existentes e o número de relacionamentos possíveis. A densidade é uma propriedade importante para avaliar o nível de coesão existente na rede.

Por exemplo, na Figura 2.7, a mesma rede, composta por 14 nós, é apresentada com número de arestas e, conseqüentemente, densidades diferentes. No diagrama (a), os nós estão interligados por apenas 14 arestas, o que resulta em uma densidade baixa de 0,15. No diagrama (b), 37 arestas relacionam os nós, representando uma densidade de 0,4. Por fim, no diagrama (c), os mesmos 14 nós ligam-se por meio de 91 arestas, onde a densidade resultante é máxima e igual a 1, pois todos os nós estabelecem ligações com os demais formando um grafo completo.

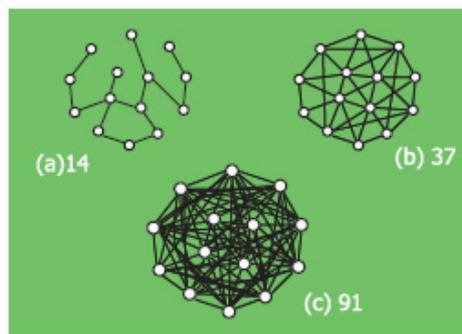


Figura 2.7 – Exemplo de densidade da rede (WWF, 2003)

2.4.4. Análises de redes sociais no desenvolvimento de software

Diversas ARS já foram feitas no desenvolvimento de software. Alguns exemplos dessas análises são descritos em (MAGDALENO *et al.*, 2010b). Muitos trabalhos exploram em especial o universo do desenvolvimento de software livre. Neste contexto, já foram estudados, por exemplo, o impacto das refatorações na coordenação de um projeto (GOTO *et al.*, 2008); e as mudanças na estrutura (centralizada, compartilhada, núcleo e periferia) da comunidade (SOUZA *et al.*, 2005).

Deve-se também observar que, além de aplicável em diferentes contextos, a ARS pode ser adotada para explorar as redes sociais sob os quatro aspectos de apoio à colaboração: comunicação (DAMIAN *et al.*, 2010, PINZGER e GALL, 2010), coordenação (SANTOS *et al.*, 2011), memória e percepção (DAMIAN *et al.*, 2010, SOUZA e REDMILES, 2011).

Neste trabalho, é importante destacar o uso da ARS para descobrir as dependências sociais entre os desenvolvedores a partir da dependência técnica entre os componentes do software. Assim, seria possível identificar os desenvolvedores que precisam se comunicar e colaborar para coordenar o trabalho. O conhecimento desta rede poderia ajudar os próprios desenvolvedores a aumentar a sua percepção e organizar mais facilmente o trabalho. Do ponto de vista do gerente de projeto, falhas na coordenação podem levar a um tempo maior de desenvolvimento, ao aumento do número de defeitos e a um incremento nos custos do projeto (CATALDO *et al.*, 2006).

A ideia é que os artefatos, que de alguma forma dependem uns dos outros, indicam a necessidade de coordenação entre as pessoas responsáveis por esses artefatos. Cataldo *et al.* (2006, 2008) também trabalham neste problema, mas utilizam uma estratégia baseada em matrizes e não em redes sociais. Os autores mostram que o alinhamento entre as dependências técnicas do software e a dependência social dos desenvolvedores pode reduzir em até 32% o esforço no projeto, pois os indivíduos têm dificuldade em identificar interdependências de tarefas que não são óbvias ou explícitas e mesmo a percepção inicial degrada conforme o projeto evolui ou os atores mudam ao longo do tempo.

Seguindo a abordagem ARS, é possível extrair três tipos de redes (BIRD *et al.*, 2009b, PINZGER e GALL, 2010, SARMA *et al.*, 2009, SOUZA *et al.*, 2007, TRAINER *et al.*, 2005, VALETTO *et al.*, 2007):

- **Rede de dependência técnica:** apresenta a rede das dependências entre as classes do código do software, onde os nós representam as classes de software e uma aresta direcionada entre duas classes indica uma relação de dependência técnica (Figura 2.8). Esta rede provê o entendimento da arquitetura do software;

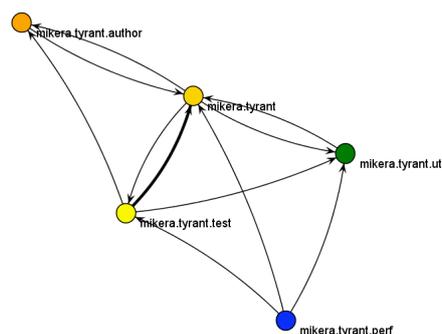


Figura 2.8 – Rede de dependências técnicas no código (TRAINER ET AL., 2005)

suporte computacional. Neste sentido, foram identificadas diversas ferramentas propostas para mineração, visualização e análise de redes sociais.

Vale ressaltar que as ferramentas tratadas aqui não são as mesmas que os famosos sites de redes sociais (BOYD e ELLISON, 2007, LAMPE *et al.*, 2009), tais como Facebook⁶. Estes sites lidam com redes sociais explícitas, onde os próprios usuários estabelecem intencionalmente os seus relacionamentos com os demais participantes. Neste trabalho, estamos interessados nas ferramentas capazes de descobrir e oferecer auxílio ao entendimento das redes sociais implícitas, ou seja, aquelas em que muitas vezes os participantes nem se dão conta de que estão integrando.

Buscando explorar as ferramentas existentes e entender as principais funcionalidades de cada uma delas, bem como identificar os principais requisitos desejáveis para explicitar a colaboração no desenvolvimento de software através das redes sociais, foram estudadas no total dez ferramentas (MAGDALENO *et al.*, 2010b, 2010c) - Ariadne⁷, Augur (SOUZA *et al.*, 2005), MiSoN (*Mining Social Networks*) (AALST *et al.*, 2005), OSSNetwork (BALIEIRO *et al.*, 2008), Pajek⁸ (NOOY *et al.*, 2005), RaisAware (COSTA *et al.*, 2010), Sargas (SOUZA JUNIOR *et al.*, 2009), SVNNAT (*Subversion Network Analysis Tool*) (SCHWIND e WEGMANN, 2008), UCINET⁹ e Visone (BRANDES e WAGNER, 2003).

As ferramentas foram classificadas em três tipos: as **ferramentas de mineração** de dados são responsáveis pelo acesso aos repositórios de dados para extrair de forma automatizada as informações necessárias para as outras ferramentas; as **ferramentas de visualização** são aquelas que oferecem funcionalidades gráficas para a compreensão visual das redes; por último, as **ferramentas de análise** de redes sociais são aquelas capazes de manipular os dados das redes sociais, através de matrizes ou grafos, para realizar os cálculos e operações necessários (Tabela 2.1).

A seguir são descritas as principais características de algumas das ferramentas estudadas. Mais detalhes sobre cada uma delas podem ser obtidos em (MAGDALENO *et al.*, 2010b).

⁶ Site do Facebook: <http://www.facebook.com>

⁷ Site da ferramenta Ariadne: <http://awareness.ics.uci.edu/~ariadne/>

⁸ Site da ferramenta Pajek: <http://pajek.imfm.si/doku.php>

⁹ Site da ferramenta UCINET: <http://www.analytictech.com/ucinet/ucinet.htm>

A OSSNetwork (BALIEIRO *et al.*, 2008) é uma ferramenta acadêmica criada por pesquisadores da UFPA (Universidade Federal do Pará) (Figura 2.11). Esta ferramenta possui mecanismos de mineração de dados para diversos tipos de fontes de informações dos projetos de software livre, como e-mail, lista de discussão, chat, repositório de gerência de configuração, sistema de controle de defeitos e código fonte.

Tabela 2.1 – Tipos das ferramentas de redes sociais

Tipo	Ferramentas
Mineração	Ariadne, Augur, MiSoN, OSSNetwork, RaisAware, SVNNAT
Visualização	Ariadne, Augur, OSSNetwork, Pajek, RaisAware, Sargas, SVNNAT, Visone
Análise	Pajek, SVNNAT, UCINET, Visone

A ferramenta Sargas aproveita a mineração de dados realizada pelo OSSNetwork e oferece mecanismos de visualização de múltiplas redes sociais. A ideia é combinar várias redes sociais para descobrir como essas diferentes redes se influenciam ou estão relacionadas umas com as outras. Entretanto, como a visualização da ferramenta mostra as redes de cada ator separadamente, perde-se a visão global da participação de todos os atores em cada uma das redes, ou seja, não existe o grafo de cada uma das redes.

A OSSNetwork (Figura 2.11) e a Ariadne possuem mecanismos de visualização temporal para redes sociais. A Ariadne trabalha com as redes de dependência técnica (Figura 2.8), sócio-técnica (Figura 2.9) e social (Figura 2.10). Contudo, apesar de estar disponível para *download*, o protótipo da ferramenta não estava funcional.

O Augur tem como objetivo prover informações da estrutura de artefatos de um projeto, de suas atividades e da rede social dos desenvolvedores envolvidos. O Augur também analisa a dependência técnica no código e monta a rede social a partir da informação de autoria, explicitando os relacionamentos entre os membros da equipe de desenvolvimento. A ferramenta oferece recursos de percepção. Entretanto, segundo os próprios autores, a ferramenta não se encontra disponível para uso e foi descontinuada.

A SVNNAT¹⁰ foi projetada para investigar a rede de colaboração entre os

¹⁰ Site da ferramenta SVNNAT: http://logatron.wiwi.uni-frankfurt.de/fileadmin/user_upload/svnnat.msi

atores do desenvolvimento de software. O objetivo da ferramenta é analisar a estrutura e explorar as propriedades da rede para melhorar o processo de desenvolvimento. Esta ferramenta, baseada na rede sócio-técnica, calcula um índice que expressa o grau de colaboração entre os desenvolvedores do projeto e extrai uma matriz de co-relação. A matriz é transformada em um grafo de rede onde os nós representam os desenvolvedores de software e os pesos nas arestas indicam o grau de colaboração entre eles. Esta ferramenta encontra-se disponível e foi utilizada no trabalho de Santos *et al.* (2010).

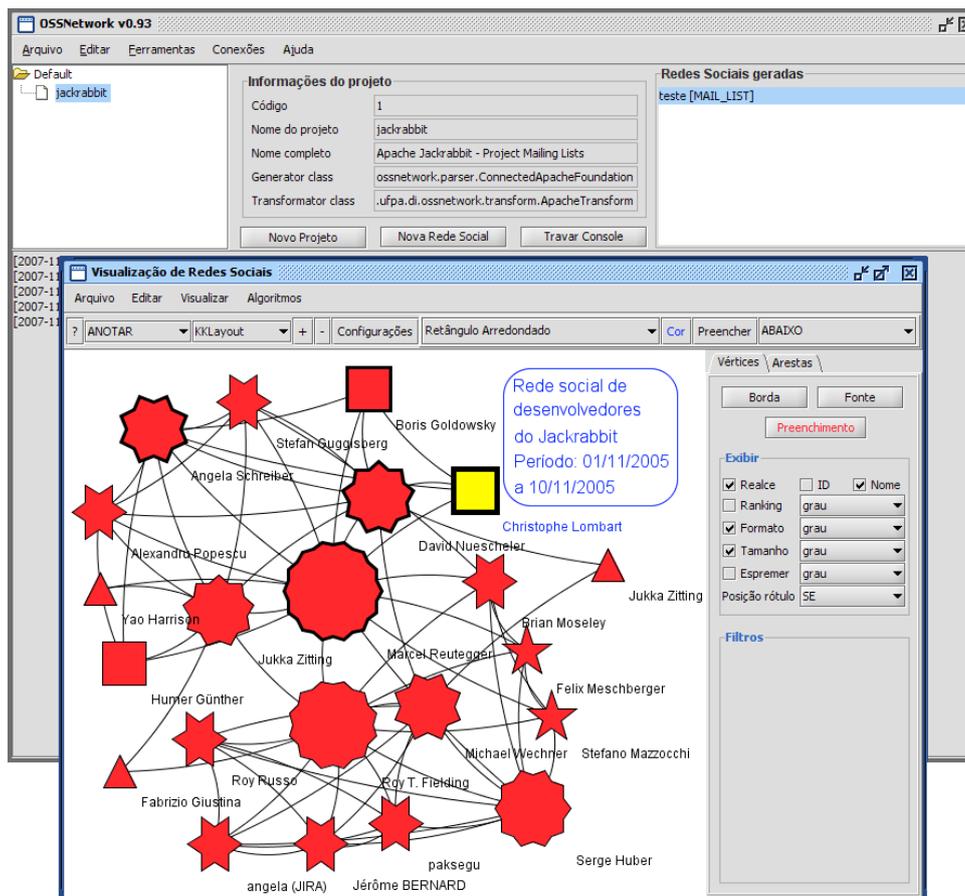


Figura 2.11 – Interface da ferramenta OSSNetwork (BALIEIRO *et al.*, 2008)

Resumindo, o estudo destas ferramentas permitiu observar que as ferramentas de análise identificadas já oferecem um conjunto extenso de algoritmos genéricos que podem ser prontamente utilizados para os cálculos das propriedades de ARS. Porém, não se dedicam a análises especificamente voltadas à colaboração. Por sua vez, a maioria das ferramentas de visualização não está de fato disponível ou possui limitações significativas. Apesar de essas ferramentas poderem visualizar diferentes tipos de rede, em geral elas não levam em consideração a evolução das redes ao longo do tempo.

2.5. Considerações Finais

De acordo com o que foi exposto neste capítulo, a colaboração é uma característica constante no desenvolvimento de software. Apesar da sua importância, uma colaboração efetiva ainda é um desafio na ES. As organizações ainda têm dificuldades em planejar e explicitar como querem que a colaboração aconteça no seu dia-a-dia de trabalho e tornar este planejamento da colaboração visível para todos os seus membros.

Foram apresentados os conceitos dos modelos de maturidade em colaboração e das redes sociais. O uso de um modelo de maturidade em colaboração serve como referência para o planejamento da colaboração em um processo. As redes sociais podem ser utilizadas para o acompanhamento da colaboração na execução de processos, pois permitem que se observe (sob diferentes perspectivas – comunicação, coordenação, memória e percepção) as interações entre os atores de um processo.

Nesta tese, investiga-se a possibilidade de planejar, otimizar e acompanhar a colaboração em processos de software. Processos são fundamentais na prática do desenvolvimento de software atualmente. A definição de um processo retrata como a interação entre os participantes é desejada, e os dados da execução de um processo retratam como a dinâmica da colaboração foi de fato exercida. Portanto, entende-se que a colaboração seja um aspecto passível de ser planejado e acompanhado nos processos definidos para distintos projetos de desenvolvimento.

3. Reutilização de Processos de Software

Neste capítulo é discutida a reutilização de processos de software como estratégia para facilitar a definição de processos de software, face a diversidade de processos em organizações. Dentre as estratégias para reutilização de processos – adaptação do processo padrão e composição de processos – a composição é adotada neste trabalho e detalhada neste capítulo. Assim, as técnicas para composição de processos – componentes e linha de processos - são apresentadas.

As organizações de software estão interessadas em aumentar a sua competitividade e produtividade para melhorar a qualidade dos produtos de software gerados e atender às necessidades dos seus clientes. Para atingir estes objetivos, pesquisas na área (CONRADI *et al.*, 1994, CUGOLA e GHEZZI, 1998, FEI DAI e TONG LI, 2007, FUGGETTA, 2000, OSTERWEIL, 1987, PAULK, 2009, RAMAN, 2000) apontam que a qualidade do produto de software depende do processo de software adotado para construí-lo e que muitos dos problemas associados ao desenvolvimento de produtos de software podem ser resolvidos aperfeiçoando-se este processo.

Um processo de software corresponde ao “conjunto de tarefas de Engenharia de Software necessárias para transformar os requisitos dos usuários em software” (HUMPHREY, 1989). Processo de software é definido por Fuggetta (2000) como: “um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos necessários para conceber, desenvolver, implantar e manter um produto de software”. A definição mais completa e objetiva vem de Paulk (2009): “o processo de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter software e seus produtos associados”.

Acredita-se que uma organização com processos de software bem definidos tem maior probabilidade de desenvolver produtos que atendam às exigências do cliente dentro do cronograma e do orçamento, quando comparada a uma organização com falhas gerenciais e sem processos definidos (VAN SOLINGEN, 2004). Portanto, o processo de software é considerado um fator crítico para o desenvolvimento de produtos de software de qualidade, uma vez que tem por objetivo gerenciar e transformar as

necessidades dos usuários em um produto de software capaz de atendê-las (ACUNA *et al.*, 2000).

Esta busca pela melhoria da qualidade tem motivado muitas organizações a investirem na definição de processos de software. A razão para se definir processos de software é melhorar a forma pela qual o trabalho é realizado. Ao pensar no processo de forma organizada, é possível antecipar problemas e antever maneiras de prevenir ou resolvê-los (HUMPHREY, 1989).

Uma das abordagens propostas para definir processos de software é prover a reutilização de processos. A reutilização de processos de software é “a capacidade de construir novos processos a partir de processos já existentes”, visto que diferentes projetos na mesma organização provavelmente terão atividades em comum (FEILER e HUMPHREY, 1992). Contar com o uso de modelos de processos já construídos como peças na construção de um novo, ao invés de construí-los a partir do zero, provavelmente irá resultar em uma construção mais eficiente do modelo (RIBÓ e FRANCH, 2002).

Em geral, tem-se reforçado a importância da reutilização de processos como forma de promover o uso de boas práticas de projetos anteriores na definição dos novos processos (ARMBRUST *et al.*, 2008, BARRETO *et al.*, 2007, ROMBACH, 2006). A reutilização de processos tem o propósito de apoiar a definição de processos com base em experiências passadas bem-sucedidas. Esta oportunidade de reutilizar a experiência adquirida durante o planejamento e condução de projetos anteriores (REIS, 2002) possibilita determinar melhores estratégias em projetos futuros (SANTOS, 2009).

Este capítulo apresenta uma visão geral dos tópicos de fundamentação teórica de reutilização de processos de software necessários ao entendimento do desenvolvimento desta tese. O restante do capítulo está organizado da seguinte forma: a Seção 3.1 mostra os desafios da diversidade de organizações, projetos, modelos de desenvolvimento de software, processos e pessoas; na Seção 3.2 são discutidas as estratégias de reutilização de processos; a Seção 3.3 é dedicada à composição de processos de software e suas principais técnicas – componentes e linha de processos; na Seção 3.4 é considerado o contexto na reutilização de processos; por fim, a Seção 3.5 conclui o capítulo.

3.1. Diversidade de Processos de Software

As iniciativas de definição de processos de software específicos para cada projeto são geralmente desafiadas pela diversidade apresentada pelas organizações, projetos, modelos de desenvolvimento, processos e pessoas.

Os contextos onde os processos definidos serão utilizados são muito diversificados. **Organizações** possuem estruturas, cultura, negócios distintos e a necessidade de diferentes tipos de processos de desenvolvimento de software. Ainda que dentro de uma mesma organização, **projetos** também podem ser diferentes, exigindo a aplicação de diferentes processos para sua execução. Ou seja, não existe um processo de software único, por mais bem definido que seja, que possa ser genericamente aplicado a todos os projetos (BERGER, 2003, CUGOLA e GHEZZI, 1998, HENDERSON-SELLERS, 2002, MACHADO, 2000, MARTÍNEZ-RUIZ *et al.*, 2008, PEDREIRA *et al.*, 2007, PEREIRA *et al.*, 2012). Dependendo das características do projeto, um processo aplicado com sucesso em um projeto pode ser um fracasso em outro. Isso torna clara a afirmação de Humphrey (1989): “Assim como não existem dois projetos de software idênticos, também não existem dois processos de software idênticos no mundo”.

Como o universo dos projetos de software é extenso, um único modelo de desenvolvimento de software¹¹ não consegue atender aos requisitos de todos os projetos, pois eles têm objetivos, características e necessidades variados (FEI DAI e TONG LI, 2007, MACHADO, 2000). Esta diversidade de **modelos de desenvolvimento de software** também traz desafios para a definição de processos de software. Os modelos de desenvolvimento de software orientado ao planejamento, ágil e livre¹² têm o mesmo objetivo: melhorar o desenvolvimento de software; mas adotam enfoques distintos.

Um projeto pode ser executado com a aplicação de diferentes **processos** (LINDVALL e RUS, 2000): (i) ao mesmo tempo dentro do mesmo projeto (diversidade *latitudinal* - observada em projetos com múltiplas equipes com realidades distintas que requerem diferentes processos de

¹¹ Um modelo de desenvolvimento de software define em alto nível de abstração um conjunto de práticas recomendadas para o desenvolvimento de software (SANTOS, 2009, SOMMERVILLE, 2004).

¹² Os detalhes sobre cada um destes modelos podem ser encontrados em (MAGDALENO, 2010b, MAGDALENO *et al.*, 2012c). A seguir são resumidas apenas algumas das suas principais características.

desenvolvimento trabalhando em paralelo); (ii) ao longo do tempo (diversidade *longitudinal* - adoção de processos diferentes ao longo do ciclo de vida de um projeto, observado, por exemplo, na transição do desenvolvimento para a fase de manutenção) (SIEBEL *et al.*, 2003).

Por fim, um processo não pode ser definido sem levar em consideração as **pessoas** que vão executá-lo (funcionários da organização) ou interagir com ele (clientes e fornecedores). É preciso conectar o processo com as pessoas e a forma como elas trabalham e interagem (SWENSON *et al.*, 2011). Ou seja, é preciso determinar qual é o processo mais adequado para uma determinada equipe ou cliente (JAKOBSEN, 2000). Porém, a diversidade de pessoas envolvidas (com diferentes motivações, experiências, conhecimentos e etc.) (DYBAA *et al.*, 2012), acrescenta mais um desafio à definição de processos de software.

Para lidar com toda esta diversidade, há uma necessidade de rápida e efetiva definição de processos de software (ALEIXO *et al.*, 2010, HANSSON *et al.*, 2006, PATEL *et al.*, 2006), que envolve o reuso de processos de software (Seção 3.2) e sua adequação ao contexto dos projetos (Seção 3.4).

3.1.1. Colaboração como um Aspecto da Diversidade de Processos de Software

A colaboração, intrínseca a qualquer processo de software, é também um aspecto impactado pela diversidade de situações de projetos e processos para seu desenvolvimento. Tomemos, por exemplo, os diferentes modelos de desenvolvimento de software praticados atualmente no mercado – orientado ao planejamento, ágil e livre.

O modelo de desenvolvimento **orientado ao planejamento** - também conhecido na literatura como tradicional, disciplinado, rigoroso ou prescritivo (BASSI, 2008) - é tipicamente exemplificado pelos modelos de maturidade e normas de qualidade, tais como o CMMI-DEV (*Capability Maturity Model Integration for Development*) (CHRISISS *et al.*, 2006), a ISO/IEC (*International Organization for Standardization*) 12207 (ISO/IEC, 1995) e o MR-MPS-SW (Modelo de Referência para Melhoria de Processos do Software Brasileiro) (SOFTEX, 2012), usados para apoiar a definição de processos de software mais previsíveis e, conseqüentemente, mais gerenciáveis.

À luz destes modelos e normas, e visando alcançar os objetivos de

previsibilidade, estabilidade e confiabilidade, este modelo se tornou fortemente orientado a planejamento e baseado em processos bem definidos e melhorados continuamente pela organização (BOEHM e TURNER, 2003).

O **desenvolvimento ágil** - ou adaptativo (BASSI, 2008) - tenta agregar valor ao negócio rapidamente e se adaptar às mudanças (de mercado, tecnologia e ambiente) que surgem durante um projeto de desenvolvimento, através de uma abordagem iterativa com ciclos de desenvolvimento mais curtos (COCKBURN, 2001, HIGHSMITH e COCKBURN, 2001, MILLER, 2001).

As características (ABRANTES e TRAVASSOS, 2007, QUMER e HENDERSON-SELLERS, 2008) e premissas do desenvolvimento ágil, observadas em métodos tais como XP (*Extreme Programming*) (BECK, 2004), Scrum (SCHWABER, 2004), FDD (*Feature Driven Development*) (PALMER e FELSING, 2002), DSDM (*Dynamic Systems Development Methodology*) (STAPLETON e CONSTABLE, 1997), *Crystal* (COCKBURN, 2004) e Kanban (ANDERSON, 2010), podem ser resumidas pelos quatro valores fundamentais do Manifesto Ágil (BECK *et al.*, 2001): **indivíduos e interações** entre eles mais que processos e ferramentas; **software em funcionamento** mais que documentação abrangente; **colaboração com o cliente** mais que negociação de contratos; **responder a mudanças** mais que seguir um plano.

Por outro lado, no modelo de desenvolvimento **de software livre**, o objetivo principal é garantir as liberdades básicas dos usuários e desenvolvedores para executar, estudar, adaptar, melhorar e distribuir o código do programa (FSF, 2008). O acesso ao código-fonte do programa é pré-requisito para estas liberdades. Entretanto, o software livre não é necessariamente gratuito e pode ser usado, desenvolvido e distribuído comercialmente (FSF, 2008).

O modelo de desenvolvimento de software livre pode ser entendido pela metáfora do *bazar* (RAYMOND, 2001), onde os projetos são desenvolvidos de forma colaborativa e transparente, sem centralização do seu planejamento e execução. Em geral, os projetos de software livre se caracterizam pelo trabalho voluntário e colaborativo de desenvolvedores, com habilidades e disponibilidades distintas, geograficamente distribuídos, organizados em uma comunidade virtual através da Internet, que contribuem voluntariamente com o desenvolvimento (CAPILUPPI *et al.*, 2003, CUBRANIC e BOOTH, 1999,

FELLER e FITZGERALD, 2001, GACEK e ARIEF, 2004, HEALY e SCHUSSMAN, 2003, REIS, 2003, WARSTA e ABRAHAMSSON, 2003).

O modelo de desenvolvimento de software livre se destaca pelo seu alto grau de colaboração e interações. Com uma equipe geograficamente dispersa, raramente seus membros se encontram pessoalmente. Devido a essa ausência de comunicação face-a-face, a comunicação acontece predominantemente de forma assíncrona e escrita, utilizando a Internet como canal de comunicação à distância (CUBRANIC e BOOTH, 1999, TAURION, 2004, YAMAUCHI *et al.*, 2000). Além disso, as comunidades contam também com a participação atuante dos usuários finais que se comunicam com os desenvolvedores e entre si, comunicando problemas e trocando experiências do uso do software (FELLER e FITZGERALD, 2001, WARSTA e ABRAHAMSSON, 2003).

A análise das principais características destes modelos permite perceber que a colaboração varia em ênfase e forma, a saber:

a) Pessoas

No desenvolvimento ágil (COCKBURN e HIGHSMITH, 2001, COCKBURN, 2001, MILLER, 2001) e livre (CUBRANIC e BOOTH, 1999), o foco está nas pessoas, nos seus talentos, habilidades e conhecimentos, e na colaboração entre elas. Os desenvolvedores de software livre são pessoas altamente motivadas e comprometidas, que se voluntariam para trabalhar no projeto. No desenvolvimento ágil, os membros da equipe também precisam ser motivados e comprometidos para lidar com a quantidade de conhecimento tácito e com as necessidades de comunicação e interação frequentes. Assim, os modelos de desenvolvimento ágil e livre são mais dependentes do talento humano do que o orientado ao planejamento, onde a explicitação do conhecimento e a padronização do trabalho diminuem esta dependência.

b) Comunicação

No modelo orientado ao planejamento, a comunicação, entre os membros da equipe de desenvolvimento e entre a equipe e o cliente, é formal e baseada em registros. O desenvolvimento ágil e o livre se baseiam na comunicação informal entre os participantes do projeto. No desenvolvimento de software livre, que opera com uma equipe geograficamente distribuída, essa comunicação informal é feita através da Internet (YAMAUCHI *et al.*, 2000). Já

no desenvolvimento ágil, orientado a equipes de desenvolvimento pequenas e fisicamente próximas, a comunicação é face-a-face (COCKBURN, 2001).

c) Coordenação

O modelo orientado ao planejamento utiliza uma estrutura de coordenação baseada no comando e controle (PRIKLADNICKI e MAGALHAES, 2010). Por outro lado, o modelo de desenvolvimento de software livre adota um modelo de liderança meritocrático, baseado no reconhecimento e reputação na comunidade, onde um ou mais indivíduos contam com a confiança e respeito do grupo para coordenar informalmente o trabalho (CUBRANIC e BOOTH, 1999, TAURION, 2004, YAMAUCHI *et al.*, 2000). Devido ao caráter de trabalho voluntário, os padrões de gerenciamento típicos não são aplicáveis. Assim, não existe uma divisão formal ou alocação prévia do trabalho, ou seja, os desenvolvedores possuem liberdade para a escolha das tarefas que serão realizadas, com base em suas preferências e habilidades (HAEFLIGER *et al.*, 2007, SCACCHI, 2007, 2010). No desenvolvimento ágil, o papel convencional do gerente de projeto como planejador, organizador e controlador é substituído por um papel de facilitador do trabalho de uma equipe que se autodireciona e se auto-organiza (BECK *et al.*, 2001, HODA *et al.*, 2010, NERUR *et al.*, 2005) para lidar com o trabalho baseado em um objetivo comum.

d) Memória

O desenvolvimento orientado ao planejamento utiliza a documentação como artefato de comunicação formal entre os membros do projeto, como instrumento para garantir a rastreabilidade dos produtos gerados entre as fases do desenvolvimento e como mecanismo de explicitação do conhecimento do projeto (BOEHM e TURNER, 2003, CHAU *et al.*, 2003). Em cada uma das fases existe uma preocupação com o registro do conhecimento explícito sobre o produto construído, a completude da documentação e a rastreabilidade entre a documentação gerada (BARBIERI, 2010, GINSBERG e QUINN, 1995, GLAZER *et al.*, 2008). Já os métodos ágeis encorajam a parcimônia na documentação (WARSTA e ABRAHAMSSON, 2003). No modelo de software livre a disponibilização livre das informações é muito importante para que os desenvolvedores possam contribuir mais eficientemente (TAURION, 2004). Assim, grande parte do conhecimento sobre o produto é tácito e compartilhado

entre os membros do projeto através da comunicação e da interação (CHAU *et al.*, 2003).

e) Relacionamento com usuários e clientes

Os usuários e clientes desempenham um papel fundamental nos três modelos de desenvolvimento, mas com atuações diferentes. No software livre, os usuários são os próprios desenvolvedores ou participam ativamente informando problemas e discutindo funcionalidades (RAYMOND, 2001).

No desenvolvimento ágil, existe um grande envolvimento e participação do cliente no levantamento, priorização e validação dos requisitos. Os usuários e clientes priorizam os requisitos em conjunto com a equipe, ficam disponíveis, tiram dúvidas, analisam as soluções propostas e fornecem *feedback* rapidamente sobre o software entregue em cada iteração (COCKBURN, 2001), ajudando assim na disseminação do conhecimento sobre o domínio de negócio para a equipe de desenvolvimento (ABRAHAMSSON *et al.*, 2003, CHAU *et al.*, 2003, COCKBURN, 2001, HIGHSMITH e COCKBURN, 2001, LINDVALL *et al.*, 2002, TURK *et al.*, 2002).

No modelo orientado ao planejamento, o cliente também desempenha um papel importante, mas somente durante as fases de especificação de requisitos e homologação (NERUR *et al.*, 2005). Além disso, o relacionamento se caracteriza por um baixo nível de confiança e é governado por definições contratuais (GLAZER *et al.*, 2008).

Tabela 3.1 – Resumo dos aspectos de colaboração nos modelos de desenvolvimento de software

Aspecto	Orientado ao planejamento	Ágil	Livre
Pessoas	Conhecimento explícito e menor dependência das pessoas	Foco nas pessoas e dependência do conhecimento tácito	Foco nas pessoas e dependência do conhecimento tácito
Comunicação	Formal e baseada em registro	Informal e face-a-face	Informal e distribuída
Coordenação	Comando e controle	Facilitação do trabalho e auto direcionamento	Meritocracia e voluntariado
Memória	Conhecimento explícito e tácito	Parcimônia e conhecimento tácito	Livre acesso e conhecimento tácito
Relacionamento com cliente	Participação em fases específicas e relacionamento baseado em restrições contratuais	Clientes chegam a ser parte da equipe de desenvolvimento ou têm participação ativa	Usuários são os próprios desenvolvedores ou têm participação ativa

3.2. Estratégias de Reutilização de Processos de Software

Reutilizar processos de software é importante para disseminar experiências de projetos anteriores (REIS, 2002); diminuir os custos e o tempo de desenvolvimento (HOLLENBACH e FRAKES, 1996); aumentar a qualidade e a adequação dos processos gerados; promover a reutilização de conhecimento de especialistas; e tornar a atividade acessível a profissionais menos experientes (BARRETO *et al.*, 2009). Xu e Ramesh (2008) complementam que modelar um processo do início, sem aproveitar processos previamente existentes, é uma atividade arriscada e trabalhosa.

Na reutilização de processos, é possível seguir duas estratégias distintas: a adaptação de um processo específico para o projeto a partir do processo padrão da organização (PEDREIRA *et al.*, 2007); ou a composição de um processo para o projeto baseada em unidades menores e reutilizáveis.

A adaptação de processos é o “ato de particularizar um processo geral para derivar uma definição aplicável a um contexto mais específico” (GINSBERG e QUINN, 1995, PEDREIRA *et al.*, 2007). Na literatura atual, observa-se uma tendência à utilização do processo padrão da organização na adaptação de processos específicos para os projetos, tal como é recomendado pelos níveis mais baixos de maturidade do CMMI-DEV (CHRISSIS *et al.*, 2006) e MR-MPS-SW (SOFTEX, 2012).

O processo padrão é um modelo abstrato que possui uma estrutura básica que guiará a definição ou adaptação de qualquer processo de software na organização, seguindo suas características e necessidades. Desta forma, o processo padrão deve garantir um conjunto de elementos fundamentais a serem incorporados em qualquer processo definido na organização (BERGER, 2003, GINSBERG e QUINN, 1995, MACHADO, 2000, ROCHA *et al.*, 2001).

Para acomodar todas as iniciativas de desenvolvimento em uma organização, o processo padrão vai inevitavelmente estar em um alto nível de abstração que atenda às necessidades de todos os projetos, mas não vai ser capaz de fornecer apoio específico às atividades individuais dos projetos (BERGER, 2003). Desta forma, o processo padrão da organização deve ser adaptado caso a caso, considerando-se as características de cada projeto.

Um processo específico para o projeto é uma coleção de atividades concretas e inter-relacionadas organizadas ao longo da linha do tempo do projeto e que leva em consideração as características do projeto (ROCHA *et al.*, 2001). O gerente de projeto é responsável por esta adaptação do processo padrão da organização para o processo específico do projeto.

Para apoiar o gerente nesta tarefa de adaptação, alguns modelos de maturidade propõem o uso de guias ou diretrizes de adaptação de processos. Estas diretrizes de adaptação são estabelecidas para guiar os projetos na seleção do modelo de ciclo de vida (dentre as opções aprovadas pela organização) e descrevem o que pode ou não ser modificado no processo específico em relação ao que constava no processo padrão.

Apesar da estratégia de adaptação ter sido bem sucedida em diversas organizações, ela não favorece a reutilização de processos (BARRETO, 2011). Para estimular a reutilização de processos, os processos também podem ser definidos com base em unidades menores e reutilizáveis de processos (CHRISISS *et al.*, 2006, SOFTEX, 2011). Neste caso, a definição de processos é realizada através da escolha dos componentes que irão compor o processo.

A composição de processos foi a estratégia adotada neste trabalho, como forma de promover a reutilização do conhecimento relacionado a processos de software. Segundo Costa (2010), definir mecanismos para auxiliar a composição de processos pode reduzir esforços de modelagem, garantir a qualidade dos processos e impedir que erros já resolvidos em projetos passados tornem a trazer problemas em novos projetos.

3.3. Composição de Processos de Software

Na composição de processos de software, algumas técnicas originalmente provenientes da reutilização de software, tais como a utilização de componentes e linhas de produtos, têm sido aplicadas e são apresentadas a seguir (KELLNER, 1996, WASHIZAKI, 2006). Em suma, nestas técnicas, os processos são o artefato no qual o conceito de reutilização é aplicado.

3.3.1. Componentes de Processos

Componentes de processos (do inglês *process component* ou *process bean*) são itens reutilizáveis que podem ser conectados para criar processos novos e

mais complexos (BARRETO, 2007, FIORINI, 2001, MAGDALENO *et al.*, 2007b). Há diversas definições para componentes de processos. Segundo Fiorini (2001), os componentes de processos consistem em soluções para problemas recorrentes relacionados com processos. Um componente de processo também pode ser visto como um encapsulamento de informações e comportamento de processo em um dado nível de granularidade (GARY e LINDQUIST, 1999).

Para o SPEM (*Software & Systems Process Engineering Meta-Model*) (OMG, 2008), um componente de processo é um agrupamento de elementos de processo que é internamente consistente e pode ser reutilizado com outros componentes de processo para compor um processo completo. Deve ser autocontido, ou seja, não deve haver referências de dentro de um componente para elementos que estão fora do componente.

Park *et al.* (2006) sugerem que um componente de processo tenha quatro interfaces: (i) os artefatos de entrada que definem os *inputs* necessários para realizar a tarefa; (ii) os artefatos de saída que definem os produtos gerados pela tarefa; (iii) uma interface de opção que indica se um componente é mandatório ou opcional; (iv) uma interface com os parâmetros de adaptação do processo que explicam a qual contexto esse componente é aplicável.

Um componente é considerado algo relevante para ser reutilizado em outras definições de processo e encapsula uma série de informações de processo, tais como: atividades, produtos de trabalho requeridos e produzidos, responsáveis pela execução, entre outros (NUNES, 2011a).

Barreto *et al.* (2011) mostram que os componentes de processo são descritos e estruturados de diferentes formas por diferentes autores, não existindo um consenso sobre quais informações devem estar contidas em um componente nem sobre o nível de detalhe a ser utilizado. A granularidade do componente depende do uso pretendido por cada abordagem.

Nunes (2011) propôs a estrutura apresentada na Tabela 3.2 para a definição de um componente de processo de software. Utilizando esta estrutura, a autora também definiu um conjunto de componentes de processo especificamente voltados para a aquisição de software.

Embora o uso de componentes para composição de processos forneça muitas vantagens, partir de unidades tão pequenas para compor grandes

processos parece ainda ser insuficiente. Conforme apontam as experiências na área de reutilização de software, o uso de estruturas reutilizáveis maiores, como por exemplo, linhas de processos de software, potencializa a composição de processos (BARRETO, 2011), conforme descrito na próxima seção.

Tabela 3.2 – Estrutura para definição de componentes de processo de software (NUNES, 2011a)

Identificador:	<Identificador único do componente de processo >
Nome:	<Nome do componente de processo >
Descrição:	<Descrição do Componente>
Tipo de Componente:	<Concreto/Abstrato>
Critérios de Entrada:	<Descrição do critério de entrada>
Critérios de Saída:	<Descrição do critério de saída>
Responsável:	<Responsável (eis) pela execução do componente>
Participantes:	<Participante (s) na execução do componente>
Artefatos Requeridos:	<Artefatos requeridos (Parâmetros de entrada do componente)>
Artefatos Produzidos:	<Artefatos requeridos (Parâmetros de saída do componente)>
Ferramentas:	<Ferramentas utilizadas>
Definido por:	<Nome da Organização que definiu a Linha de Processo>
Características Atendidas:	<Características atendidas pela Linha de Processo>
Características Conflitantes:	<Características que quando selecionadas, impedem o uso da Linha de Processo>
É variante de	< Identificador do componente abstrato a que este componente concreto atende>

3.3.2. Linha de Processos de Software (LPS)

A ideia de Linhas de Processos de Software (LPS) (ALEIXO *et al.*, 2010, BARRETO, 2007, JAUFMAN e MUNCH, 2005, ROMBACH, 2006, TERNITE, 2009, WASHIZAKI, 2006) surge da transferência do paradigma de Linhas de Produtos (NORTHROP, 2002) para o domínio de processos.

A ideia de Linhas de Produtos tem como proposta a construção sistemática de software baseada em uma família de produtos. Um dos principais pontos é a identificação das características comuns e variáveis entre os produtos, possibilitando um aumento na flexibilidade e adaptação às necessidades dos clientes. Uma linha de produtos é um “conjunto de produtos de software que compartilham um conjunto de características comuns e controladas que satisfazem necessidades de um segmento de mercado em particular e são desenvolvidos a partir de artefatos, de forma predefinida” (NORTHROP, 2002). Uma linha de produtos funciona como uma fábrica, que instancia produtos com características similares, mas, ao mesmo tempo, com algumas características específicas que os diferenciam, através da composição de componentes existentes (GARG *et al.*, 2003, NORTHROP, 2002). Desta forma, uma linha de produtos pode ser vista como uma coleção de produtos similares (SUTTON e OSTERWEIL, 1996).

Um conceito chave para as linhas de produtos é a variabilidade, que pode ser entendida como “a habilidade que um artefato possui de ser alterado, customizado, ou configurado para um contexto em particular” (BOSCH, 2004). A variabilidade em uma linha é importante para explicitar os pontos onde tais produtos se assemelham e, portanto, podem ser reutilizados, e os pontos onde eles diferem entre si, devendo receber tratamento específico. Desta forma, a variabilidade oferece flexibilidade para diferenciar e diversificar os produtos.

A ideia da variabilidade também está presente no contexto de processos. Um processo de software quase nunca é executado da mesma forma duas vezes. Organizações, projetos, equipes sempre introduzem diversidade e variabilidade em um processo ao usá-lo, quando se inicia um novo projeto. Por outro lado, existem também alguns aspectos do processo que se mantêm inalterados, podendo ser considerados comuns ao processo ao longo de muitas execuções.

As LPS são linhas de produtos cujos produtos são processos. Washizaki (2006) define uma LPS como “um conjunto de processos de um determinado domínio de problema ou com um determinado propósito, que tem características em comum e é construído baseado em ativos reutilizáveis de processos”. Segundo Ternite (2009) uma linha de processos é definida de forma similar: “conjunto de processos que capturam similaridades e variabilidades controladas. Cada um desses processos é desenvolvido a partir de um conjunto comum de ativos reutilizáveis (características) de forma predefinida”. Sutton e Osterweil (1996) sugeriram o conceito de “família de processos”, que significa basicamente o mesmo que uma linha de processos, mas sem a conotação de uma sequência linear. Barreto *et al.* (2009) definem uma linha de processos como “uma arquitetura de processos que estabelece os elementos (componentes ou atividades) principais dos processos e como eles se relacionam”. Neste trabalho adota-se a seguinte definição elaborada pelo grupo de pesquisa no contexto do Projeto CDSOft (MAGDALENO *et al.*, 2012a, NUNES *et al.*, 2010b, TEIXEIRA, 2011):

Um conjunto de elementos de processos que compartilham características comuns e variáveis dentro de um domínio específico e são desenvolvidas a partir de artefatos que podem ser reutilizados e combinados entre si, segundo regras de composição e recorte, para compor e adaptar processos.

O uso de uma LPS oferece uma ampla flexibilidade na criação de diferentes processos (HENDERSON-SELLERS, 2002). Por exemplo, em uma LPS, os três modelos de desenvolvimento de software podem realmente coexistir, sendo representados através de variabilidades e opcionalidades.

Os objetivos de uso de LPS são: aumentar a produtividade da composição de processos, diminuindo o esforço necessário para realizá-la; aumentar a qualidade e adequação dos processos gerados; representar variabilidades e semelhanças entre processos para potencializar a reutilização; e diminuir os riscos de uma composição inadequada (BARRETO *et al.*, 2009, JAUFMAN e MUNCH, 2005, PEDREIRA *et al.*, 2007, ROMBACH, 2006).

3.3.2.1. Propostas para Linha de Processos de Software

Esta seção resume as principais propostas para LPS. No trabalho de Teixeira (2011) estas propostas são descritas em mais detalhes e analisadas comparativamente segundo alguns requisitos de representação de variabilidades.

Na abordagem de Barreto *et al.* (2009), que tem foco na alta maturidade, a LPS definida é utilizada para derivar instâncias de processos a partir da seleção de um conjunto de características. Para povoar essa LPS, os autores adotam tanto a estratégia *bottom-up* para componentizar processos legados, definidos e executados nas organizações (BARRETO *et al.*, 2009, 2011) quanto a estratégia *top-down* para a identificação de componentes para o processo de aquisição de software. A Figura 3.1 retrata um exemplo de uma LPS especificamente voltada para o processo de aquisição de software, pois a proposta prevê a criação de múltiplas LPS específicas para cada processo padrão da organização e não de uma única LPS completa. Em um trabalho complementar, Nunes (2011) sugere um formulário com os principais atributos para a definição de uma LPS.

O trabalho de (BARRETO, 2011) desenvolveu ainda **ferramentas de apoio à definição de processos baseada em reutilização** que permitem criar a LPS através da definição de características e componentes de processos. Esses elementos são armazenados em um repositório de componentes. A LPS criada pode ser utilizada para definir o processo padrão da organização e os processos específicos para os projetos.

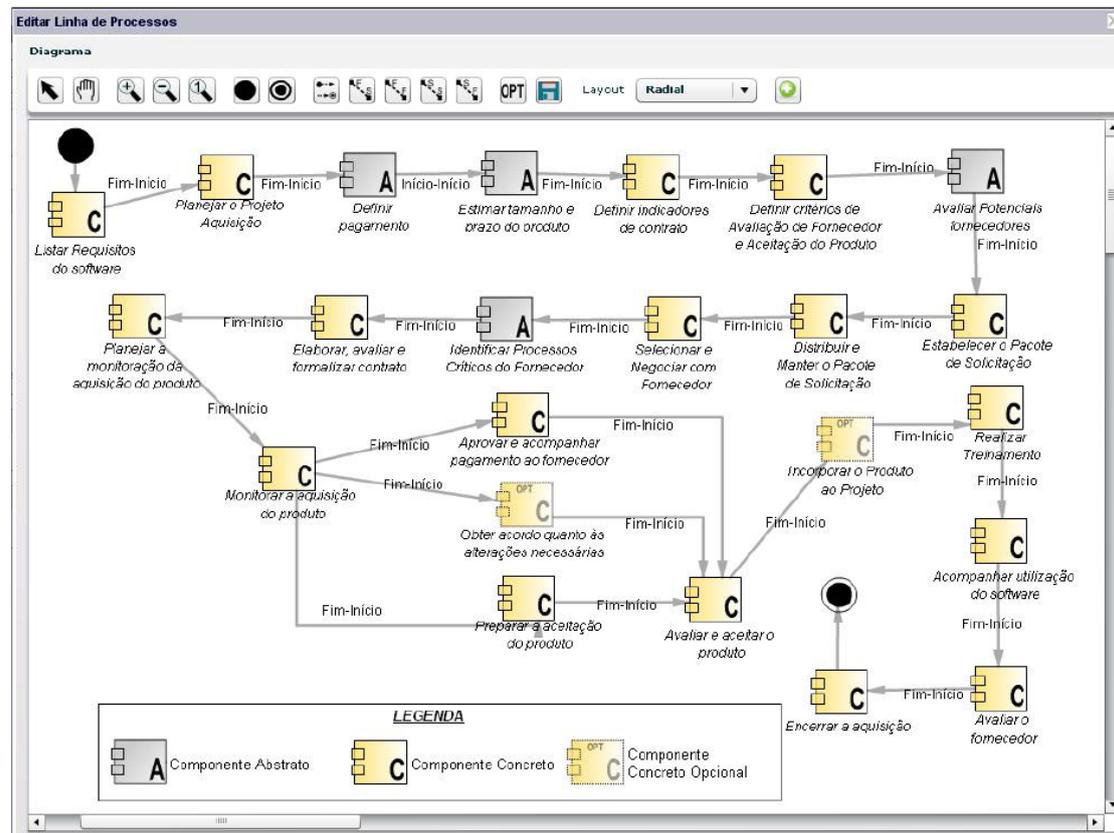


Figura 3.1 – Linha de Processos de Aquisição de Software (NUNES et al., 2010a)

Rombach (2006) propôs a Engenharia de Linha de Processos de Software (ELPS), cujo objetivo é, por meio de um processo de engenharia de domínio, criar um conjunto de processos genérico que capturasse as semelhanças e variabilidades em um domínio. A engenharia de linhas de processo inclui: o processo de *engenharia de domínio*, que cria processos para reutilização; o processo de *engenharia de aplicação*, que cria processos específicos para um projeto; e um *repositório de artefatos* que disponibiliza processos reutilizáveis em todos os níveis de abstração. Esta necessidade de uma abordagem sistemática e que separe as partes comuns e variáveis do processo também é defendida por Armbrust et al. (2008).

Washizaki (2006) define a ELPS como “um sistema de estratégias inter-relacionadas e abordagens sistemáticas para construir, aplicar e gerenciar a linha de processos”. O autor propõe uma abordagem *bottom-up* para estabelecer arquiteturas de linhas de processos a partir dos processos existentes, que incorporam semelhanças e variabilidades (Figura 3.2). Nesta proposta, as características do projeto também são levadas em consideração.

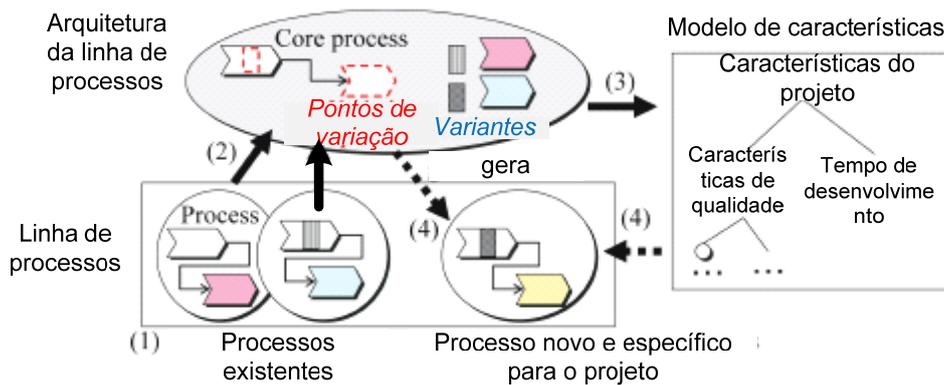


Figura 3.2 – Abordagem *bottom-up* para linha de processos (WASHIZAKI, 2006)

A proposta de Aleixo *et al.* (2010a) avalia as técnicas existentes para a gestão de variabilidades em linhas de processos de software. A especificação de processo é anotada com variabilidades, as quais são refletidas em um modelo de características e um modelo de configuração, que armazena o mapeamento de características específicas para elementos de processo. Usando técnicas de transformação de modelos, os autores advogam que os processos customizados podem ser automaticamente transformados em especificações para execução em uma máquina de workflow.

A proposta, especificamente voltada para linha de processos de software, inclui as seguintes etapas: (i) determinar os membros da linha de processos; (ii) utilizar um processo de engenharia de domínio para construir um repositório de processos; (iii) recortar a linha de processos extraíndo uma instância de processo; e (iv) customizar/adaptar a instância do processo. Esta abordagem trata tanto da criação da LPS quanto do recorte de processos a partir da linha.

A análise desses trabalhos ressalta a importância de se considerar tanto a criação da LPS quanto o recorte de processos a partir da linha. Além disso, a maioria das propostas de LPS existentes na literatura trata a modelagem no nível do projeto, modelando arquiteturas de componentes de processo. No entanto, a modelagem de variabilidades pode ser representada em um modelo de mais alto nível de abstração, como um modelo de características.

Por fim, apenas a proposta de Aleixo *et al.* (2010a) sugere algum tipo de automação para apoiar a derivação de processos específicos para os projetos. Porém, esta proposta está voltada para uma abordagem de transformação de modelos que não é o mesmo caminho de pesquisa deste trabalho.

3.4. Reutilização de Processos de acordo com Contexto

Para lidar com a diversidade de processos, também é preciso levar em consideração o contexto onde os processos serão aplicados. Esta seção resume alguns trabalhos relacionados que utilizam as particularidades do contexto do projeto para a reutilização de processos.

a) Contingency factors e situational method engineering

Situational method engineering (SME) (BRINKKEMPER, 1996, SLOOTEN e BRINKKEMPER, 1993) propõe a construção de métodos de desenvolvimento de software para situações específicas (HENDERSON-SELLERS e RALYTÉ, 2010, PEREIRA, 2012). Esta área tem por objetivo oferecer flexibilidade às organizações na configuração de um processo específico para o projeto de desenvolvimento, usando métodos ou fragmentos de métodos, construídos pela própria organização, armazenados em um repositório e recuperados para satisfazer a situação de um projeto em particular (HARMSEN *et al.*, 1994, HENDERSON-SELLERS e RALYTÉ, 2010).

SME define os seguintes quatro passos para a construção de um método (ou definição de um processo específico para o projeto): (i) especificação da situação do projeto; (ii) definição dos requisitos dos fragmentos de método, que refletem as situações; (iii) seleção de um conjunto de fragmentos que satisfaça os requisitos; (iv) construção de uma metodologia coerente a partir da montagem dos fragmentos selecionados (ABAD *et al.*, 2010).

A área de *contingency factors* (BEKKERS *et al.*, 2008, KUMAR e WELKE, 1992) refere-se às variáveis que caracterizam um projeto para determinar a seleção do método apropriado a partir de um portfólio de métodos (DAVIS, 1982). Assim, SME e *contingency factors* podem ser combinados para recuperar fragmentos de acordo com as características situacionais, que auxiliam na determinação de quais elementos são mais apropriados. Essa ideia de caracterização do projeto é semelhante ao proposto neste trabalho, em que as informações de contexto são usadas para compor um processo que utilize as características e componentes de uma LPS.

Uma das principais críticas feitas a essas abordagens é que há uma

carência de orientações sobre o modo como os *contingency factors* se relacionam com os métodos de desenvolvimento (KUMAR e WELKE, 1992). Além disso, não são encontrados relatos de experiências práticas da aplicação dessas abordagens e existe uma carência por dados resultantes de experimentos na área (HENDERSON-SELLERS e RALYTÉ, 2010).

Outro problema fundamental é que o gerente de projeto precisa de apoio para tomar decisões sobre o processo a ser adotado no projeto. Enquanto SME pode apresentar todos os detalhes sobre cada fragmento de processo individual, não dá indicações sobre a seleção e combinação deles para garantir que o processo resultante é executável. Esta capacidade de compor este processo, considerando o contexto do projeto, é uma contribuição da presente tese.

No contexto da área de pesquisa de SME, Pereira (2012) propôs a abordagem *Octopus SME – Process Tailoring Approach* (OSPTA). Trata-se de uma sistemática para construção de processos de software específicos para cada projeto, através da adaptação de processos, com a utilização de fragmentos de métodos (PEREIRA, 2012, PEREIRA *et al.*, 2012).

A abordagem OSPTA visa à construção de um processo específico para cada projeto, que atenda as características situacionais do mesmo. Assume que já existe um processo padrão da organização definido. A Figura 3.3 apresenta uma visão geral da abordagem OSPTA, detalhando as suas principais atividades.

Primeiramente, devem ser definidas as características situacionais do projeto. Então, o projeto deve ser analisado em relação ao(s) critério(s) de adaptação estabelecido(s). Os fragmentos de métodos adequados em relação a um ou mais critérios de adaptação são recuperados da base de métodos de acordo com o resultado da atividade anterior. Para isso, deve haver na base de métodos fragmentos previamente associados ao critério de adaptação. Caso não existam, o responsável deve editar um fragmento existente ou definir novos elementos e armazená-los na base de métodos.

A ferramenta de apoio, usando um algoritmo de priorização, classifica os fragmentos previamente recuperados em ordem de relevância em relação ao contexto do projeto para sugerir os fragmentos mais apropriados para serem inclusos no processo de software adaptado. Após a priorização dos fragmentos

pelo algoritmo, o engenheiro de processos pode selecionar o conjunto final de fragmentos de métodos. Finalmente, o processo adaptado é construído pela ferramenta de apoio. Estudos de caso considerando “riscos do projeto” como critério de adaptação foram conduzidos para validar a abordagem (PEREIRA, 2012, PEREIRA *et al.*, 2012).

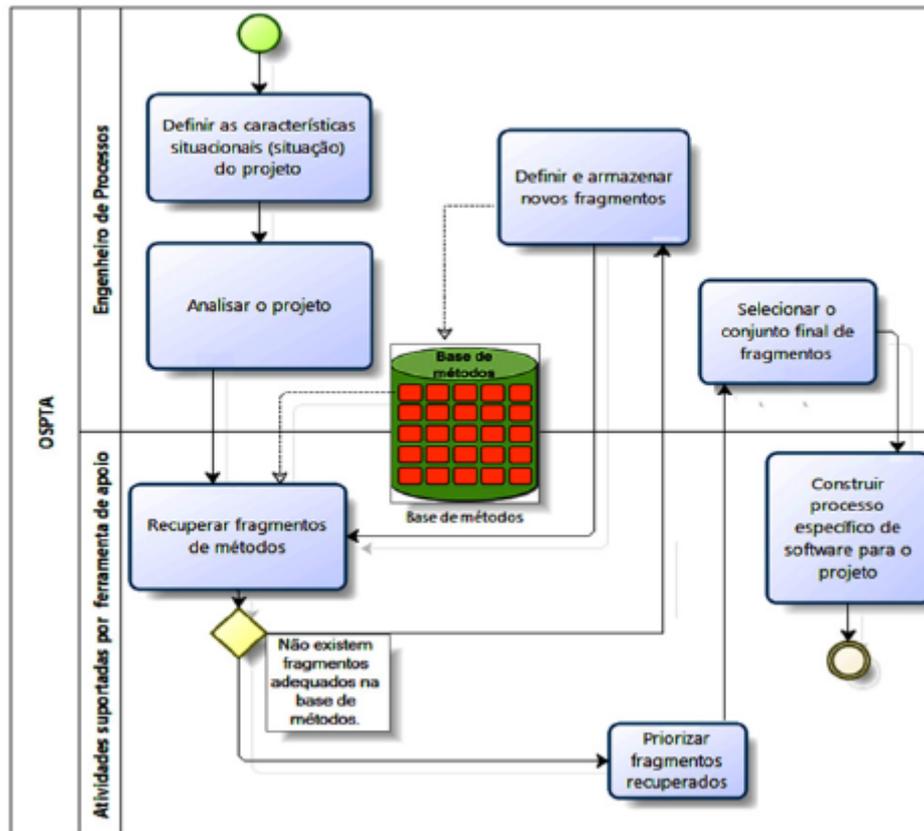


Figura 3.3 – Visão Geral da Abordagem OSPTA (PEREIRA, 2012)

b) Gerenciamento dinâmico de ativos de processos de software

A abordagem para gerenciamento dinâmico de ativos de processos de software (Figura 3.4) tem o foco na aprendizagem organizacional (SANTOS, 2009). O componente principal da abordagem é o repositório de ativos de processos, onde são armazenados os modelos de processo de software reutilizáveis. Estes modelos são contextualizados através de atributos de projeto e processo.

O usuário fornece os requisitos de processo através dos atributos de projeto e processo, e a máquina de busca realiza o cálculo de similaridade para os casos do repositório. A máquina de busca utiliza a técnica de Raciocínio Baseado em Casos (RBC) para a recuperação de casos similares através da medida de similaridade da representação de cada caso. Assim, esta

abordagem requer a existência de uma base de casos, ou seja, um conjunto de processos que já foram adaptados para contextos de projetos específicos.

Após a recuperação e a adaptação do(s) processo(s), conforme Figura 3.4, um processo é definido para atender às necessidades de um determinado projeto da organização. Ao encerrar a execução do processo, a instância do processo reutilizado é avaliada em relação à sua utilidade. Assim, é possível realimentar o repositório da base de casos de processos. O registro do fracasso ou do sucesso ajuda a melhorar o resultado das futuras situações.

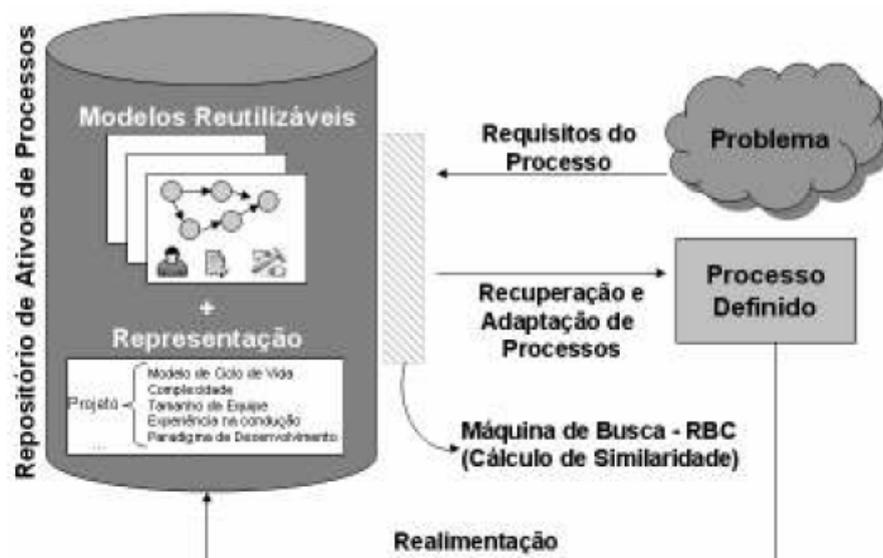


Figura 3.4 – Abordagem para gerenciamento dinâmico de ativos de processos de software (SANTOS, 2009)

Dependendo da avaliação realizada sobre o processo executado, o usuário poderá escolher entre generalizá-lo ou não. O ato de generalizar consiste em transformar o processo executado em um modelo de processo abstrato, isto significa que os detalhes específicos do processo são retirados para deixá-lo apropriado para a reutilização em diversos projetos.

3.5. Considerações Finais

Este capítulo complementou o referencial teórico desta tese. Os processos de software foram abordados sob a perspectiva da diversidade de organizações, projetos, modelos de desenvolvimento de software, processos e pessoas.

Como abordagem para a definição de processos de software considerando a diversidade, foram apresentadas as estratégias de reutilização de processos de software. A composição de processos é a estratégia adotada

neste trabalho, como forma de promover a reutilização do conhecimento relacionado a processos de software. Por isso, técnicas de composição, tais como o uso de componentes e linhas de processos, foram abordadas no capítulo.

Um aspecto particular da diversidade de processos, a colaboração, foi discutido. Observa-se que, independente do modelo de desenvolvimento utilizado, a colaboração está presente no processo, com diferentes ênfases e formas. Esta observação reforça a questão introduzida no capítulo anterior, de que a colaboração possa ser planejada na definição do processo para um projeto e posteriormente acompanhada em sua execução, como forma de abordar a diversidade apresentada por diferentes contextos e situações.

4. Projeto CDSOft

Neste capítulo é descrito o contexto de desenvolvimento deste trabalho de pesquisa - o Projeto CDSOft. São apresentados a visão geral, os objetivos do projeto e as diferentes estratégias de solução aplicadas pelo projeto para o problema da composição de processos de software. O escopo desta tese é então delimitado no contexto do Projeto CDSOft e em relação aos demais trabalhos de pesquisa em desenvolvimento pelo projeto.

Este trabalho de pesquisa está inserido no contexto do Projeto CDSOft¹³ (WERNER *et al.*, 2011a), que tem como objetivo construir soluções para o apoio à composição de processos de software de forma sistemática e dinâmica. O projeto foi iniciado em 2010, motivado pelas ideias e resultados parciais da presente tese de doutorado. Os objetivos gerais desta pesquisa de doutorado permitiram delinear o arcabouço geral do projeto e definir o foco de atenção e escopo específicos desta tese neste contexto.

Para enfrentar um desafio que envolve diferentes áreas de conhecimento, o Projeto CDSOft reúne o Grupo de Reutilização de Software do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ e o Programa de Pós-Graduação em Informática (PPGI) da UNIRIO. O projeto é realizado por um grupo formado por: docentes de ambas as instituições; três alunos de doutorado, sendo um deles a autora desta tese; e alunos de mestrado e graduação, alocados em trabalhos de pesquisa específicos ao projeto.

Neste capítulo, são apresentados a visão geral, os objetivos do projeto e as diferentes estratégias de solução aplicadas no CDSOft para o problema da composição de processos de software: a Engenharia de Linha de Processos de Software (ELPS); a gestão de informação de contexto; e a combinação destas duas estratégias em um método para Linha de Processos baseada em Contexto (LPBC). O objetivo específico desta tese no contexto do projeto CDSOft é então apresentado, cujo escopo está no planejamento, acompanhamento e otimização do aspecto da colaboração na composição de processos de software, tendo como base a LPBC.

O capítulo está organizado da seguinte forma: a Seção 4.1 apresenta a

¹³ Site: <http://reuse.cos.ufrj.br/cdsoft/>

visão geral do Projeto CDSOft. A Seção 4.2 é dedicada à abordagem ELPS de reutilização de processos de software. A gestão de contexto, incluindo a estrutura de representação de contexto e um exemplo de modelo de contexto para o domínio de processos de software, é o foco da Seção 4.3. A abordagem LPBC proposta no projeto para a combinação de linha de processos e gestão de contexto é detalhada na Seção 4.4. Na Seção 4.5, é resumido o papel da otimização na composição de processos de software. A Seção 4.6 delimita o escopo da solução proposta nesta tese, dentro do contexto do Projeto CDSOft. Por fim, a Seção 0 tece considerações finais ao capítulo.

4.1. Visão Geral do Projeto CDSOft

O Projeto CDSOft (Figura 4.1) visa propor uma sistemática e apoio computacional para a automação dos passos da composição de processos de software, diminuindo o esforço necessário para sua execução e melhorando os resultados obtidos.

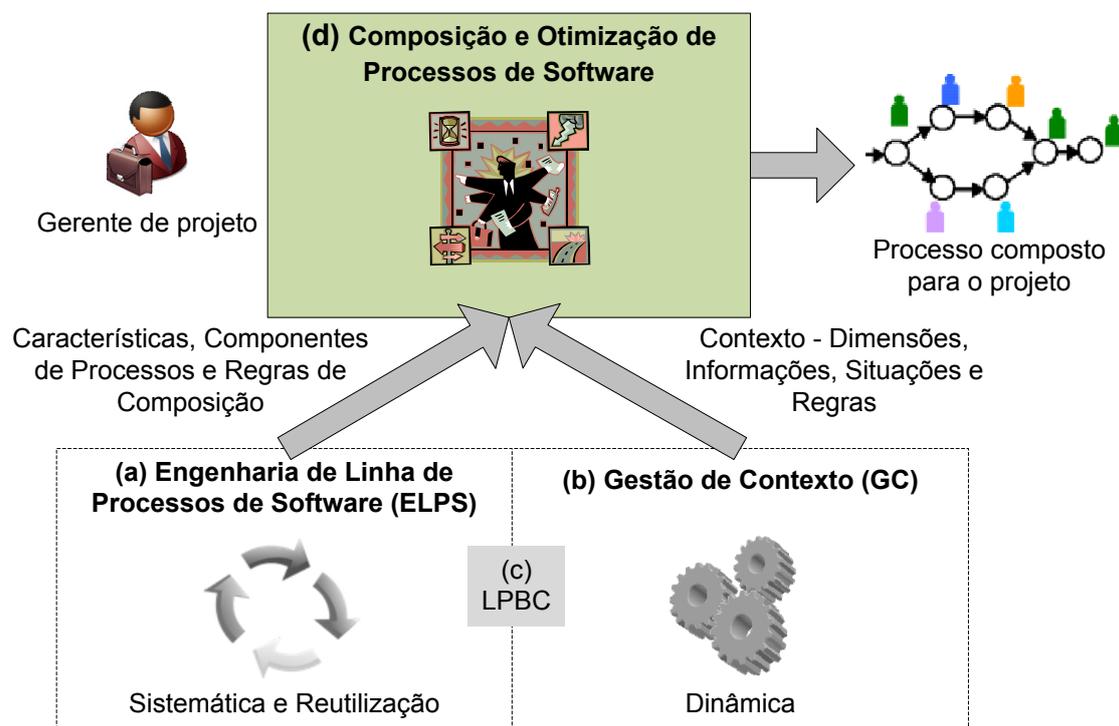


Figura 4.1 – Visão geral da solução para composição de processos no Projeto CDSOft

A sistemática proposta pelo projeto tem o objetivo de guiar a composição de processos, visando caracterizar, usar e gerenciar as similaridades e diferenças entre os processos. A sistemática é estabelecida valendo-se de uma

estrutura de reutilização de processos baseada em Linha de Processos de Software (LPS), conforme visto no Capítulo 3.

Assim, no Projeto CDSOft e, mais especificamente, na dissertação de Teixeira (2011), foi construída a abordagem de **Engenharia de Linha de Processos de Software (ELPS)** (Figura 4.1a). Essa abordagem consiste na especificação das etapas para criação e uso de uma LPS e seus respectivos produtos. Na ELPS foram definidos quatro elementos principais: um meta-modelo, uma notação, um método e um ferramental de apoio, descritos na Seção 4.2.

A estrutura de reutilização de processos foi combinada com soluções de **Gestão de Contexto (GC)** (NUNES, 2011b) (Figura 4.1b) visando: estabelecer relações entre uma determinada situação de contexto e as características e os componentes da LPS; compreender o contexto atual de um projeto através de um conjunto de informações de contexto; e identificar mudanças no contexto da organização, do projeto e da equipe ao longo da execução do projeto. A GC visa permitir que as sugestões de composição do processo possam ser revistas e continuamente atualizadas, desde o planejamento à execução do processo.

A combinação da ELPS e GC deu origem à abordagem de **Linha de Processos Baseada em Contexto (LPBC)** (Figura 4.1c) (MAGDALENO *et al.*, 2012a, NUNES *et al.*, 2010b), detalhada na Seção 4.4. A LPBC oferece a habilidade de adaptar o comportamento do processo às mudanças no contexto. De forma flexível, ela permite ativar ou desativar alternativas de componentes de processos na linha dependendo do contexto do projeto. Isso permite que o recorte do processo satisfaça às mudanças nas necessidades do projeto.

A LPBC fornece insumos para a **composição e otimização de processos** (Figura 4.1d) (MAGDALENO *et al.*, 2012a). Estes insumos são conjuntos de dados sobre o processo e o contexto do projeto, da organização e da equipe. Os dados recebidos são combinados e analisados para sugerir ao gerente de projeto alternativas para a composição do processo para o projeto. As sugestões apresentadas otimizam aspectos específicos de interesse da organização, satisfazendo as restrições de contexto.

A visão geral da solução proposta pelo projeto para a composição de processos de software é apresentada na Figura 4.1. Apesar de existirem propostas que, de alguma forma, tratam alguns destes elementos isoladamente

(ALEIXO *et al.*, 2010, BARRETO, 2011, MARTÍNEZ-RUIZ *et al.*, 2011, ROMBACH, 2006, ROSEMAN e RECKER, 2006, SAIDANI e NURCAN, 2007, SANTORO *et al.*, 2007, WASHIZAKI, 2006), o Projeto CDSOft se diferencia ao propor sua integração na direção de viabilizar a construção de um ambiente integrado para a composição, acompanhamento e otimização de processos de software.

Dentro do escopo do Projeto CDSOft, a COMPOOTIM (MAGDALENO *et al.*, 2012a) é o foco desta tese, sendo detalhada na Seção 4.6. A COMPOOTIM está voltada para a maximização da colaboração nos processos compostos para os projetos de software. A COMPOOTIM inclui uma sistemática e um ferramental de apoio para oferecer uma solução baseada na otimização da colaboração na composição de processos de software.

4.2. Engenharia de Linha de Processos de Software (ELPS)

A ELPS é a proposta do Projeto CDSOft para estruturar a reutilização de processos de software e permitir que os processos de software também possam ser organizados de acordo com as suas similaridades e diferenças, facilitando e guiando o seu uso de acordo com as necessidades específicas dos projetos. Para alcançar este objetivo, a ELPS é formada por quatro elementos: método, meta-modelo, notação e ferramental de apoio (Figura 4.2). Estes elementos são brevemente descritos nas próximas seções. Informações adicionais podem ser obtidas em (TEIXEIRA, 2011).

4.2.1. Meta-Modelo e Notação

Para atender aspectos específicos de uma modelagem de características dentro da área de reutilização de processos, foi necessária a construção de um meta-modelo que representasse, explicitamente, os conceitos de características, variabilidade e opcionalidade associados à representação dos elementos que constituem a definição de processos de software, como atividades, tarefas, papéis e produtos de trabalho.

Assim, foi proposto o meta-modelo *OdysseyProcess-FEX* (TEIXEIRA, 2011), inspirado no SPEM (*Software Process Engineering Metamodel*) (OMG, 2008) e baseado na *Odyssey-FEX* (OLIVEIRA, 2006), utilizada para representação de variabilidades através de modelos de características para

linhas de produtos de software. Desta forma, o meta-modelo *OdysseyProcess-FEX* também é baseado na modelagem de características, apresentada na próxima seção.



Figura 4.2 – Elementos da ELPS (TEIXEIRA, 2011)

4.2.1.1. Modelagem de Características

A modelagem de características é uma das técnicas mais aceitas para representar produtos em uma linha de produtos (LEE *et al.*, 2002). O propósito da modelagem de características é “capturar e gerenciar as similaridades e diferenças, de forma a facilitar o entendimento de clientes e desenvolvedores no que se refere às capacidades gerais de um domínio, que são expressas através de características” (KANG *et al.*, 1990).

Uma *característica de processo* é uma adaptação do conceito usado tradicionalmente em linhas de produtos para o domínio de processos. As características são tipicamente organizadas em um modelo de características, comumente usado para representar variabilidade nas linhas.

O conceito de **variabilidade** é trabalhado através da classificação de uma característica como *Ponto de Variação*, *Variante* ou *Invariante*. Essa classificação quanto à variabilidade é mutuamente excludente, ou seja, cada característica não pode ser classificada com mais de um dos tipos de variabilidade definidos:

- **Pontos de variação:** refletem a parametrização no domínio de uma maneira abstrata e podem ser configurados através das variantes, ou seja, podem ser realizados por diferentes variantes;

- **Variantes:** atuam como alternativas para se configurar um ponto de variação, ou seja, uma variante é uma das possíveis implementações de um ponto de variação;
- **Invariantes:** características fixas que representam elementos não configuráveis em um domínio.

Estes conceitos são apresentados no exemplo ilustrado pela Figura 4.3, no domínio de Processos de Engenharia de Requisitos, onde a característica “Elicitar Requisitos” é um ponto de variação, que possui como variantes as características “Realizar Entrevistas”, “Aplicar Questionários”, “Usar Cenários”, “Realizar *Brainstorming*” e “Construir Protótipos”.

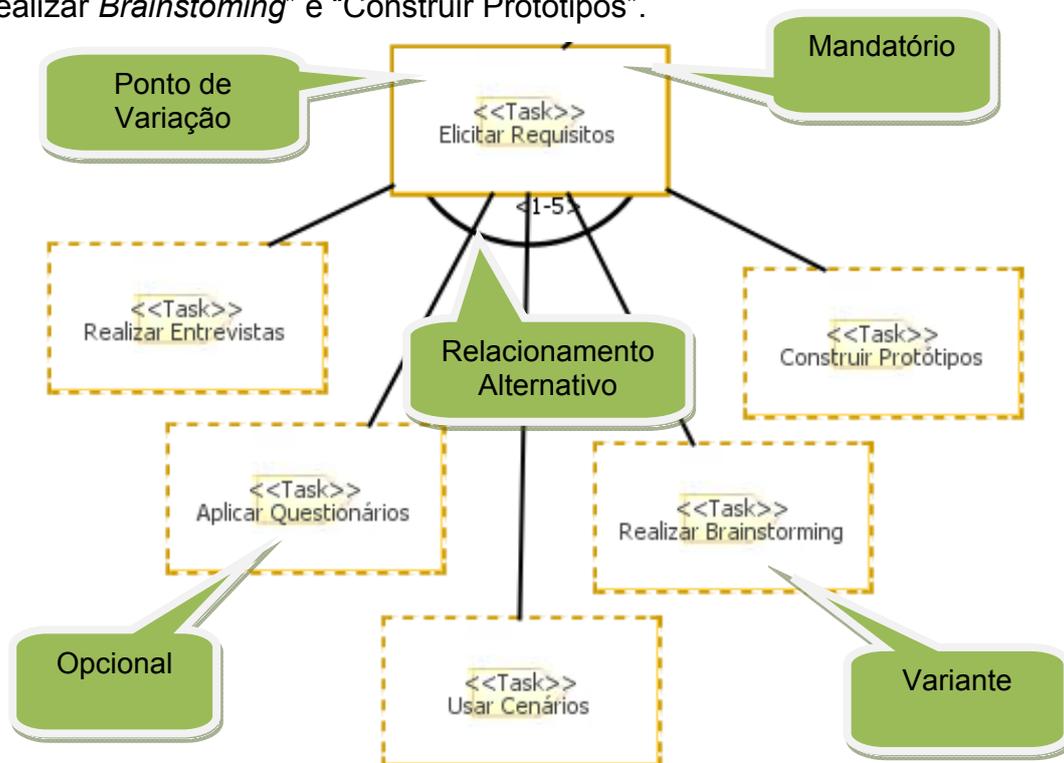


Figura 4.3 – Exemplo de modelo de características com variabilidade

Em relação à **opcionalidade**, as características podem ser classificadas como (BLOIS, 2006, LINDEN *et al.*, 2007, OLIVEIRA, 2006):

- **Elementos opcionais:** elementos que podem ou não estar presentes nos processos derivados a partir da LPS;
- **Elementos mandatórios:** elementos que devem obrigatoriamente estar presentes em todos os processos derivados a partir da LPS.

No exemplo da Figura 4.3, a opcionalidade é retratada nas características através de uma linha pontilhada e a obrigatoriedade através da linha contínua. Logo, com exceção da característica “Elicitar Requisitos”, as

demais características podem ou não ser incluídas no processo composto para o projeto ao recortar a LPS.

O meta-modelo *OdysseyProcess-FEX* combina relacionamentos próprios do modelo de características (como o alternativo) com relacionamentos da UML (herança, composição, agregação e associação). Um exemplo do relacionamento alternativo também é apresentado na Figura 4.3.

Outro conceito relacionado à modelagem de características é a **cardinalidade**, que é utilizada para definir o número mínimo e máximo de características que podem ser escolhidas a partir de um conjunto de alternativas de um ponto de variação. No exemplo da Figura 4.3, é possível observar que uma ou mais técnicas de “Elicitar Requisitos” (cardinalidade é <1-5>) podem ser selecionadas.

A especificação de **restrições** entre as características, que inclui os relacionamentos de dependência e mútua exclusividade, é uma forma de indicar a necessidade ou incompatibilidade da seleção conjunta de características (OLIVEIRA, 2006). Tais conceitos podem ser representados através do uso de regras de composição:

- **Inclusivas:** definem relações de dependência entre duas ou mais características, indicando que elas devem ser selecionadas em conjunto;
- **Exclusivas:** definem relações de mútua exclusividade entre duas ou mais características que não devem ser escolhidas em conjunto.

As regras de composição são expressas pela seguinte estrutura: antecedente + palavra-chave + conseqüente. A palavra-chave representa o tipo de regra: “requer” (*requires*), referente às regras inclusivas; e “exclui” (*excludes*), referente às regras exclusivas. Antecedente e conseqüente são expressões (literais ou *booleanas*) e representam uma característica ou combinação de características.

É importante destacar que o conceito de característica de processo tem sido utilizado com semântica diferente em outras abordagens. No trabalho de Barreto (2011), uma característica pode ser vista como um aspecto, qualidade ou caracterização com a qual o processo precisa ser compatível e com semântica definida pelo usuário. Características de processo são usadas apenas como um mecanismo de alto nível para seleção de componentes.

4.2.1.2. *OdysseyProcess-FEX*

O meta-modelo *OdysseyProcess-FEX* (TEIXEIRA, 2011) visa explicitar a representação de conceitos de elementos de processos e de variabilidades do domínio. Possui um conjunto de restrições e propriedades que juntas constituem as regras de boa formação do modelo. Essas regras direcionam a construção e a verificação de consistência de um modelo de características do domínio de processos de software. Além disso, define o conjunto de relacionamentos disponíveis entre os conceitos representados.

Junto ao meta-modelo é apresentada a notação *OdysseyProcess-FEX*, que tem como objetivo representar simbolicamente os conceitos formalizados pelo meta-modelo. Cada característica de processo pode ser classificada segundo sete categorias. Para cada categoria definida foi especificado um símbolo e um estereótipo associado à representação visual dos conceitos descritos (Tabela 4.1).

Tabela 4.1 – Categorias das características na notação *OdysseyProcess-FEX* (TEIXEIRA, 2011)

Categoria		Símbolo
Disciplina	Característica que representa uma categorização de trabalho, relacionado com uma grande área de interesse no âmbito do domínio como um todo.	 «discipline»
Atividade	Característica que representa o agrupamento de unidades de trabalho menores, representadas por outras atividades ou por tarefas.	 «activity»
Tarefa	Característica que representa uma unidade fundamental de trabalho.	 «task»
Prática	Característica que representa uma forma ou estratégia comprovada de realizar um trabalho.	 «practice»
Conceito	Característica que descreve ideias-chave, aborda temas gerais ou princípios básicos.	 «concept»
Papel	Característica que representa um conjunto de habilidades, competências e responsabilidades de indivíduo(s).	 «role»
Produto de Trabalho	Característica que representa um artefato consumido, modificado ou produzido por uma tarefa.	 «work product»

Um exemplo com algumas das disciplinas do RUP (*Rational Unified Process*) (IBM, 2009) é apresentado na Figura 4.4. Quase todas, à exceção da “Implementação” e “Teste”, foram definidas como opcionais para que seja possível no momento do recorte do processo selecionar apenas aquelas que

interessam para o projeto em questão. As disciplinas “Implementação” e “Teste” foram definidas como obrigatória por ser considerada essencial para qualquer projeto da organização. Elas não poderão ser excluídas durante a composição do processo.

Uma vez identificadas as disciplinas, é possível identificar as suas principais atividades. As atividades correspondem a um agrupamento lógico de unidades menores de trabalho, que podem corresponder a outras Atividades ou a Tarefas. Alguns exemplos de atividades foram modelados no diagrama da Figura 4.4. No caso da atividade “Estimativa” foram identificadas três possíveis tarefas dedicadas às estimativas de tamanho, esforço e custo.

Do ponto de vista da representação de processos de software, vale destacar que uma limitação significativa do meta-modelo *Odyssey-ProcessFEX* (TEIXEIRA, 2011) originalmente proposto é que ele não aborda o sequenciamento/fluxo dos processos, tipicamente presente na modelagem de processos. Porém, este meta-modelo atualmente encontra-se em revisão no escopo de outra tese de doutorado do grupo que visa detalhá-lo justamente com foco na fase de projeto de domínio de processos e no seu principal artefato - os componentes de processos. Enquanto este trabalho está em andamento, no Projeto CDSOft optou-se por adotar uma estrutura simples (Tabela 4.2) para a definição dos componentes, ou seja, sem considerar todos os elementos de processo, para dar prosseguimento aos demais trabalhos do grupo, inclusive a presente tese de doutorado.

Tabela 4.2 – Estrutura para definição de componentes de processo de software

Campo	Descrição
Identificador:	<Identificador único do componente de processo>
Nome:	<Nome do componente de processo>
Papel:	<Papel responsável pela execução do componente de processo>
Descrição:	<Descrição do componente de processo>
Característica:	<Característica de processo a qual o componente de processo está associado>
Opcionalidade:	<Opcionalidade do componente de processo – <i>Mandatório ou Opcional</i> >
Variabilidade:	<Variabilidade do componente de processo - <i>Ponto de Variação, Variante ou Invariante</i> >
Interfaces Requeridas:	<Artefatos requeridos pelo componente>
Interfaces Providas:	<Artefatos produzidos pelo componente>
Potencial de Colaboração:	<Potencial de colaboração do componente de processo>

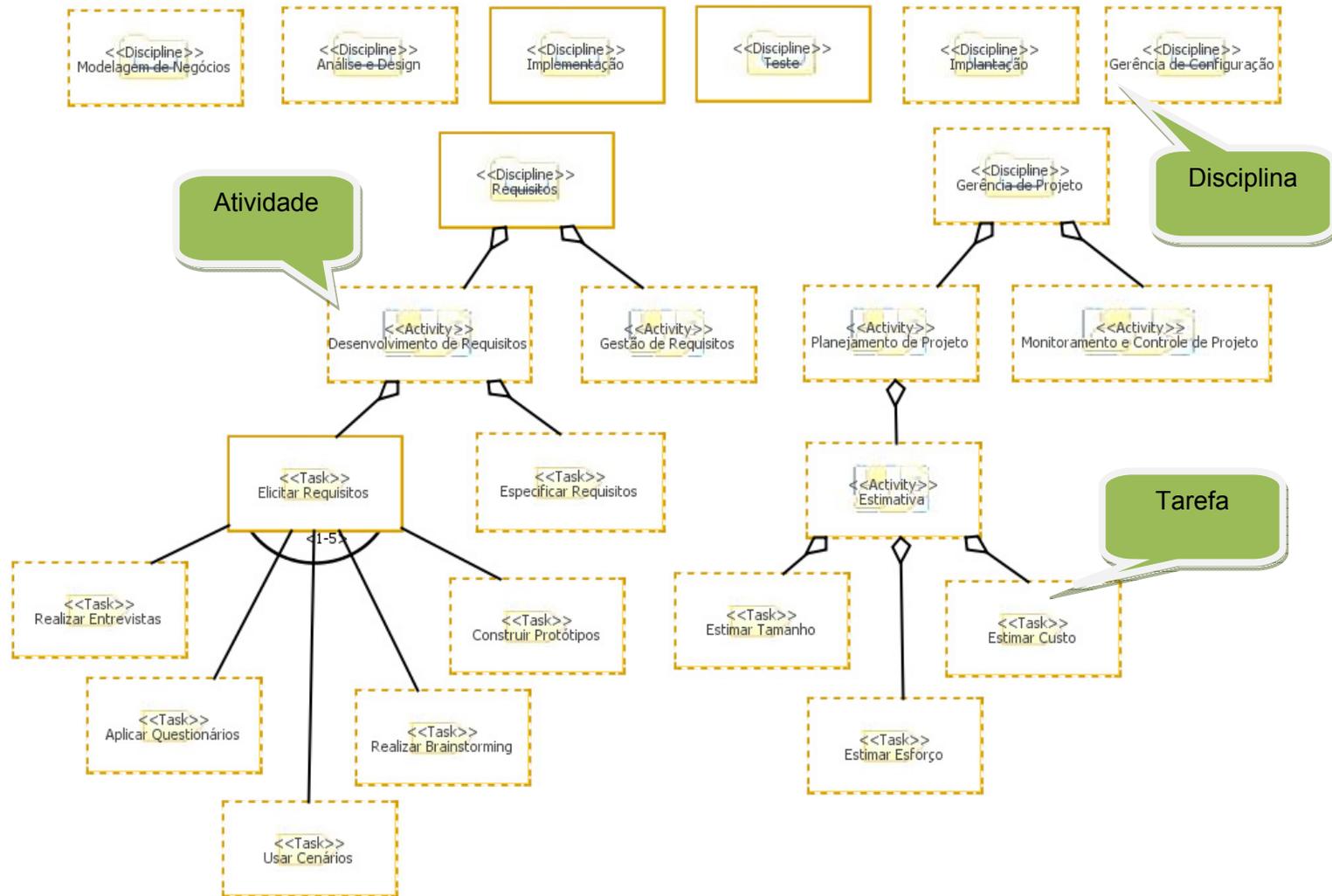


Figura 4.4 – Exemplo parcial de modelo de características de processo na notação Odyssey-ProcessFEX

4.2.2. Método

O método define os passos para capturar os elementos comuns e variantes dos processos de forma a criar definições de processos que possam ser aplicadas em uma variedade de situações, ou seja, que sejam reutilizáveis. Assim, o método estabelece a sequência de passos para a criação da LPS e para o recorte de processos a partir da linha. Os detalhes do método serão apresentados na Seção 4.4.

Para a aplicação do método são necessários alguns insumos, providos por estratégias de captura de conhecimento apresentadas a seguir. Existem duas estratégias de captura de conhecimento possíveis para a ELPS: *bottom-up* e *top-down* (BARRETO *et al.*, 2011, ROMBACH, 2006, WASHIZAKI, 2006), que se diferenciam pelos insumos utilizados.

Na estratégia *bottom-up*, o conhecimento existente nos modelos de processos da organização ou nas instâncias executadas dos projetos de desenvolvimento de software é recuperado através de análise ou mineração dos processos (*process mining*) (AALST e GIINTHER, 2007, AALST *et al.*, 2004), e utilizado para criar a LPS.

Por sua vez, a estratégia *top-down* reflete a necessidade de padronização de processos. Nesta estratégia, inicia-se a linha a partir de modelos de referência, onde processos ou partes deles são adquiridos de forma a serem combinados para formar novos processos. Esta estratégia é mais adequada quando não existe um conjunto de processos previamente definidos na organização e os requisitos/objetivos para a criação da LPS estão claros (BARRETO *et al.*, 2010).

4.2.3. Ferramental de apoio

No Projeto CDSOFT, o ambiente Odyssey (ODYSSEY, 2013) foi adotado como ferramental de apoio para a representação da LPS conforme especificado pelo meta-modelo e pela notação e para que dê suporte às diferentes atividades definidas pelo método (Figura 4.2).

O ambiente Odyssey, desenvolvido pelo Grupo de Reutilização de Software da COPPE/UFRJ, é uma infraestrutura de reutilização baseada em modelos de domínio e contempla atividades de Engenharia de Domínio (ED) e Engenharia de Aplicação (EA). O Odyssey permite que se tenha, de forma

unificada no mesmo ambiente, o suporte para a modelagem do domínio, através do modelo de características (Figura 4.3), e para a modelagem contextual (que será apresentada na próxima seção).

Apesar de esta infraestrutura ter sido concebida com foco em produtos e não em processos, ela já oferece diversos mecanismos para apoiar cada uma das fases da abordagem, ainda que alguns deles precisem de adaptações ou extensões. A evolução do ambiente Odyssey para o domínio de processos de software foi iniciada por Teixeira (2011), contemplando a etapa de análise de domínio. As demais etapas da ED têm tido continuidade no escopo da sua pesquisa de doutorado, que também é parte integrante do Projeto CDSOft.

4.3. Gestão de Contexto (GC)

Uma definição comumente adotada declara que contexto é “qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade, onde uma entidade pode ser uma pessoa, lugar, ação ou objeto que seja considerado relevante para a situação” (DEY *et al.*, 2001). No Projeto CDSOft, contexto é definido segundo Brezillon (1999): “contexto é uma descrição complexa do conhecimento compartilhado sobre circunstâncias físicas, sociais, históricas e outras dentro das quais ações ou eventos ocorrem”.

Na combinação de contexto com processos, o contexto é definido como: “o mínimo de variáveis contendo informações relevantes que impactem a definição e execução do processo” (ROSEMANN e RECKER, 2006, SAIDANI e NURCAN, 2007). Na Engenharia de Software, o contexto também é importante, pois não é de se esperar que uma tecnologia (ou processo) seja universalmente boa ou ruim, apenas mais apropriada em algumas circunstâncias (DYBAA *et al.*, 2012).

Para que o contexto possa ser realmente útil na prática, ele precisa ser gerido. Uma infraestrutura para gestão de contexto em processos foi proposta por Nunes *et al.* (2010c) e inclui a captura, armazenamento, recuperação e *reasoning* do contexto, baseados em um modelo ou estrutura de representação de contexto (NUNES, 2011b, NUNES *et al.*, 2011, 2012).

4.3.1. Estrutura de representação de contexto

No Projeto CDSOft optou-se por utilizar uma estrutura de representação de contexto também baseada em modelos de características. Assim, foi adotada a notação UbiFEX (FERNANDES, 2008, FERNANDES *et al.*, 2008), que estende a notação Odyssey-FEX (OLIVEIRA, 2006), incluindo de forma explícita elementos para a modelagem de contexto de um determinado domínio.

A visão geral da abordagem UbiFEX, composta por quatro atividades, é apresentada na Figura 4.5. A primeira atividade é definir as entidades de contexto. As entidades de contexto representam as dimensões de contexto relevantes para o domínio.

Em seguida, devem ser definidas as informações de contexto que devem ser coletadas para caracterizar as entidades de contexto do domínio. Tanto as entidades quanto as informações de contexto podem ser classificadas em mandatórias ou opcionais e possuem um conjunto de atributos específicos. Para a representação explícita das entidades e informações de contexto, duas categorias de características foram definidas, com o mesmo nome dos elementos que representam (Tabela 4.3).

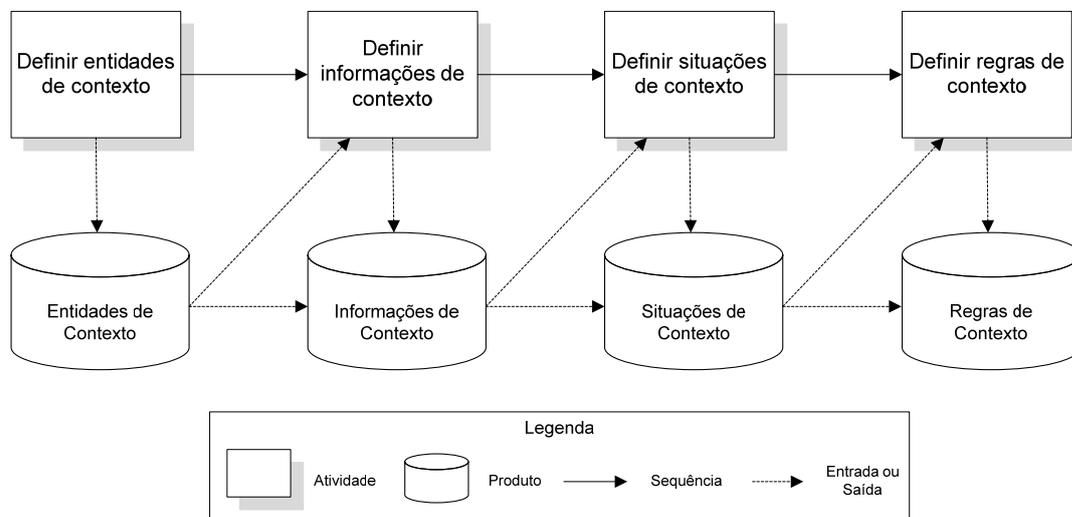


Figura 4.5 – Visão geral da abordagem UbiFEX

O terceiro passo é caracterizar as situações de contexto que podem acontecer e que acionam as ações de configuração do processo de uma LPS. As definições de contexto descrevem cenários relevantes para o domínio, tendo como base as entidades e informações de contexto modeladas anteriormente. A UbiFEX permite que uma nova situação de contexto seja definida a partir de outras previamente definidas, promovendo o reuso e

facilitando a atividade de manutenção. Os elementos que compõem a expressão de uma situação de contexto estão formalmente descritos em (FERNANDES, 2008).

Por fim, as ações que devem ser tomadas quando uma determinada situação acontece podem ser especificadas pelas regras de contexto. Estas regras definem como uma situação de contexto impacta na configuração de um processo, indicando, por exemplo, decisões a respeito da seleção de variantes em um ponto de variação do modelo de características. Os elementos que compõem a estrutura de uma regra de contexto e as restrições para definição destas regras estão formalmente descritos em (FERNANDES, 2008).

Tabela 4.3 – Características de contexto (FERNANDES, 2008)

	Ícone	Categoria	Descrição
Características de contexto		Entidade de Contexto	Características que representam as entidades de contexto relevantes para o domínio.
		Informação de Contexto	Características que representam as informações que devem ser coletadas para caracterizar as entidades de contexto do domínio.

A modelagem de contexto, usando a notação UbiFEX, também foi implementada no ambiente de reutilização Odyssey (ODYSSEY, 2013) e assim já conta com apoio computacional. Um exemplo de aplicação desta estrutura de representação de contexto no domínio de processos de software é apresentado na próxima seção.

4.3.2. Modelo de Contexto para Processos de Software

Esta seção introduz o modelo de contexto para processo de software que vem sendo usado pelo Projeto CDSOft. Este modelo segue a estrutura de representação de contexto da UbiFEX (FERNANDES *et al.*, 2008), apresentada na seção anterior. O modelo foi apresentado inicialmente em (MAGDALENO, 2010b) e, posteriormente, foi revisto, formalizado e validado no escopo da dissertação de mestrado de Leite (2011).

4.3.2.1. Dimensões de contexto

Seguindo a abordagem UbiFEX, o primeiro passo para construção de um modelo de contexto é definir as entidades ou dimensões de contexto para o domínio de processos de software. A proposta de Araujo *et al.* (2004) fornece

uma visão das principais dimensões de contexto envolvidas no desenvolvimento de software: *indivíduo, papel, equipe, tarefa, projeto, organização, domínio da engenharia de software, produto de software, domínio do negócio e cliente/usuário* (Figura 4.6).

Ainda que todas as dimensões agreguem informações de contexto importantes, as dimensões *organização, projeto* e *equipe* são particularmente relevantes no escopo do Projeto CDSOft. As dimensões *organização* e *projeto* foram consideradas, visto que a composição de processos, geralmente, envolve estas duas dimensões (PEDREIRA *et al.*, 2007). A dimensão *equipe* também é importante, dado o foco deste trabalho de pesquisa no aspecto da colaboração.

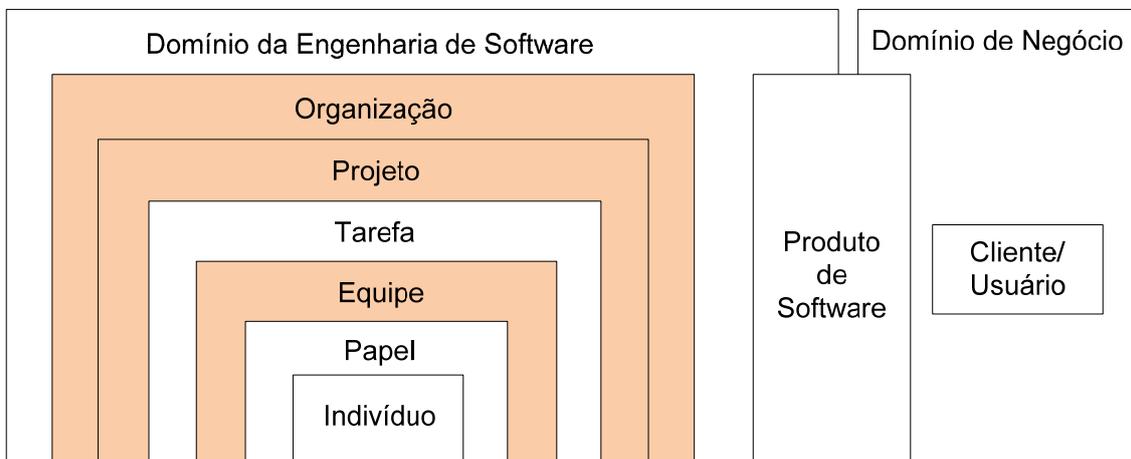


Figura 4.6 – Dimensões de contexto no desenvolvimento de software (ARAUJO *et al.*, 2004)

4.3.2.2. Informações de contexto

Para cada uma das dimensões de contexto, pode-se pensar em um conjunto de informações de contexto úteis para a composição de processos de software. A determinação das informações de contexto não é uma atividade trivial (ROSEMANN e RECKER, 2006, SAIDANI e NURCAN, 2007). Vários tipos de informações contribuem com o contexto e a relevância de cada informação, depende do foco de atenção ou da tarefa em questão (BREZILLON, 1999).

Um conjunto inicial de informações de contexto relevantes para caracterizar um projeto no domínio de processos de software foi obtido através da revisão e adaptação de trabalhos já publicados na literatura (ARMBRUST *et al.*, 2008, BEKKERS *et al.*, 2008, BERGER, 2003, BOEHM e TURNER, 2003, BURNS e DENNIS, 1985, COCKBURN, 2000, 2001, GINSBERG e QUINN, 1995, HENDERSON-SELLERS, 2002, LAANTI e KETTUNEN, 2005, LITTLE,

2005, MACHADO, 2000, MNKANDLA, 2008, PARK *et al.*, 2006, PEREIRA, 2012, PEREIRA *et al.*, 2012, QUMER e HENDERSON-SELLERS, 2008, SANTOS, 2009, SLOOTEN e BRINKKEMPER, 1993, WEERD, 2009, XU e RAMESH, 2008) ou através da própria experiência da autora deste trabalho em definição de processos de software.

Além disso, foi realizada por Leite (2011) uma revisão sistemática da literatura para ajudar a identificar critérios comumente adotados na adaptação de processos de software. Este resultado foi validado com os especialistas da prática e da comunidade científica com o objetivo de avaliar a pertinência e relevância das informações identificadas. O resultado deste levantamento foi incorporado ao modelo de contexto para processos de software.

Alguns exemplos de informações de contexto para a dimensão equipe são apresentados na Tabela 4.4. Estas informações de contexto foram modeladas no Odyssey e são apresentadas na Figura 4.7. O conjunto completo de informações de contexto, bem como as suas definições, pode ser consultado em (ARAUJO *et al.*, 2012, LEITE, 2011, MAGDALENO *et al.*, 2011b).

Tabela 4.4 – Exemplos de informações de contexto para processos de software

Dimensão	Informações de Contexto	Composição	Valores
Equipe	Tamanho da equipe	<i>Atômica</i>	- Muito grande – > 80 pessoas; - Grande – 40 a 80 pessoas; - Média – 11 a 39 pessoas; - Pequena – 7 a 10 pessoas; - Muito pequena – < 6 pessoas
	Experiência técnica	Experiência no domínio de aplicação	- Muito alta – > 6 anos; - Alta – 4 a 6 anos; - Média – 2 a 4 anos; - Baixa – 1 a 2 anos; - Muito baixa – < 1 ano
		Experiência no desenvolvimento de software	- Muito alta – > 6 anos; - Alta – 4 a 6 anos; - Média – 2 a 4 anos; - Baixa – 1 a 2 anos; - Muito baixa – < 1 ano
		Experiência na tecnologia	- Muito alta – > 6 anos; - Alta – 4 a 6 anos; - Média – 2 a 4 anos; - Baixa – 1 a 2 anos; - Muito baixa – < 1 ano
	Experiência gerencial	<i>Atômica</i>	- Muito alta - especialista; - Alta - muito experiente; - Média - experiente; - Baixa - pouco experiente; - Muito baixa - sem experiência
	Experiência no trabalho em equipe	<i>Atômica</i>	- Muito alta; - Alta; - Média;

Dimensão	Informações de Contexto	Composição	Valores
			- Baixa; - Muito baixa
	Proximidade	<i>Atômica</i>	- Localizada – equipe reunida no mesmo local de trabalho; - Distribuída – equipe dispersa em diferentes localizações
	Estabilidade	<i>Atômica</i>	- Muito alta; - Alta; - Média; - Baixa; - Muito baixa

A dificuldade em definir as informações de contexto relevantes para um determinado domínio aponta para a necessidade de uma sistemática que guie este trabalho. Assim, no Projeto CDSOft, o método ORGANON foi desenvolvido por Anastassiou (2012) para identificar informações de contexto de processos de quaisquer domínios. Porém, a presente definição das informações de contexto para processos de software foi feita antes da criação deste método e, portanto, ainda requer uma revisão e maior formalismo.

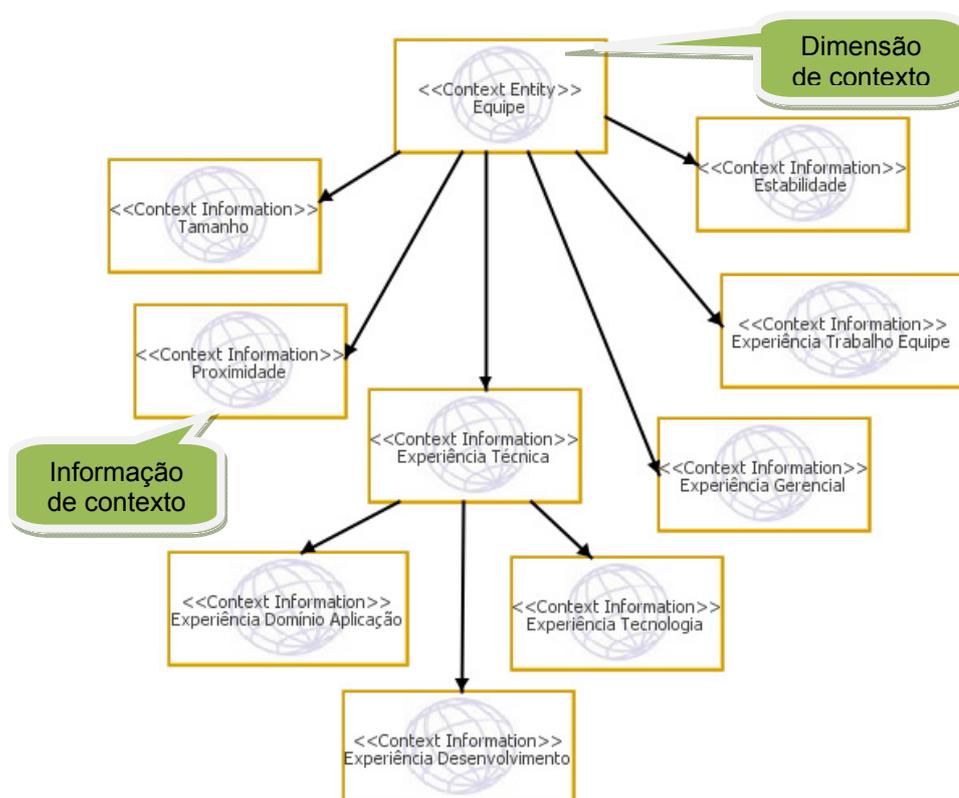


Figura 4.7 – Exemplo parcial de dimensões e informações de contexto no Odyssey

4.3.2.3. Situações de contexto

Para o uso das informações de contexto dentro do domínio de processos de software, é necessário estabelecer situações que impactem no

estabelecimento de regras ou sugestões de combinações de práticas de desenvolvimento de software. As situações de contexto representam circunstâncias que podem acontecer baseadas na combinação de valores de determinadas informações de contexto. Para ilustrar, são apresentadas na Tabela 4.5 algumas situações da dimensão equipe propostas por Leite (2011).

- *Equipe Inexperiente* – equipe com pouca ou nenhuma experiência;
- *Equipe em Aprendizado* – equipe que possui algum nível de experiência;
- *Equipe Experiente* – equipe com alta experiência;
- *Equipe Próxima* – equipe fisicamente concentrada em um único local;
- *Equipe Dispersa* – equipe geograficamente distribuída.

Tabela 4.5 – Exemplos de situações de contexto de equipe

Informação de Contexto	Equipe Inexperiente	Equipe em Aprendizado	Equipe Experiente	Equipe Próxima	Equipe Dispersa
Experiência no Domínio de Aplicação	Muito Baixa / Baixa	Média	Muito Alta / Alta	-	-
Experiência no Desenvolvimento de Software	Muito Baixa / Baixa	Média	Muito Alta / Alta	-	-
Experiência na Tecnologia	Muito Baixa / Baixa	Média	Muito Alta / Alta	-	-
Experiência no Trabalho em Equipe	Muito Baixa / Baixa	Média	Muito Alta / Alta	-	-
Proximidade da Equipe	-	-	-	Localizada	Distribuída

4.3.2.4. Regras de contexto

Após a definição das situações de contexto relevantes para o domínio, as ações que devem ser tomadas caso uma determinada situação ocorra podem ser especificadas por meio das regras de contexto. Para ajudar a composição do processo, é necessário estabelecer regras de contexto que relacionem as situações de contexto à seleção de características da LPS. As regras de contexto restringem a escolha das características de acordo com o contexto.

Alguns exemplos de regras de contexto foram construídos utilizando parte das situações de contexto listadas na seção anterior e o exemplo de modelo de características da LPS (apresentado na Seção 4.2). A Tabela 4.6 resume estes exemplos de regras de contexto.

Tabela 4.6 – Exemplos de regras de contexto

# regra	Situação de Contexto	Operador	Característica
CR1	Equipe Experiente AND Equipe Próxima	<i>Implies</i>	Realizar Entrevistas
CR2	Equipe Dispersa	<i>Implies</i>	Aplicar Questionários

4.4. Linha de Processos Baseada em Contexto (LPBC)

Não basta que a composição de processos seja feita uma única vez no início do projeto, pois os requisitos e o ambiente do projeto mudam. Portanto, a composição de processos deve ser executada de forma contínua, considerando o contexto de execução do processo e mantendo o alinhamento do processo com as necessidades atuais do projeto (BRINKKEMPER, 1996, FITZGERALD *et al.*, 2003, MAGDALENO, 2010a, XU e RAMESH, 2008). Porém o entendimento sobre como o contexto afeta a composição de processos é limitado (XU e RAMESH, 2008) e ainda não existe suporte tecnológico eficiente (BARRETO, 2011, JAUFMAN e MUNCH, 2005).

Assim, foi proposta por (NUNES *et al.*, 2010b) a utilização de informações de contexto para apoiar a composição de processo e melhorar a utilização e gestão da LPS. Combinando a sistemática das LPS com a dinâmica provida pela GC, a abordagem LPBC tem como objetivo construir, utilizar e gerir uma LPS (Figura 4.8).

Com o uso da abordagem LPBC, espera-se que ao recortar um processo a partir da linha, o esforço requerido para a composição deste processo seja diminuído (WASHIZAKI, 2006). A pesquisa de Hollenbach e Frakes (1996) mostra que é possível diminuir em pelo menos dez vezes o tempo e o esforço necessário para criar o processo de um projeto quando se aproveita um processo reutilizável, ao invés de criá-lo desde o início.

De forma análoga à engenharia de linha de produtos, a abordagem LPBC também compreende duas grandes fases (MONTERO *et al.*, 2007, ROMBACH, 2006, SCHNIEDERS e PUHLMANN, 2006): Engenharia de Domínio de Processos (EDP) e Engenharia de Aplicação de Processos (EAP) (Figura 4.8). Na EDP, o Grupo de Definição de Processos de Software (GDPS) ou o equivalente existente na organização cria processos para reutilização e é

responsável por definir as similaridades e variabilidades da linha. Na EAP, o Gerente de Projeto reutiliza os artefatos gerados pela EDP, ao recortar processos específicos para um projeto a partir da linha, explorando a variabilidade da linha, levando-se em consideração as necessidades específicas do projeto. A EDP e a EAP trabalham em conjunto para promover a reutilização: a EDP cria a LPS e provê um conjunto de artefatos, tais como o modelo de características e o modelo de componentes de processo, enquanto a EAP aplica regras e compõe os processos com base no recorte dos artefatos gerados na EDP. Ao final, os processos estão prontos para serem executados.

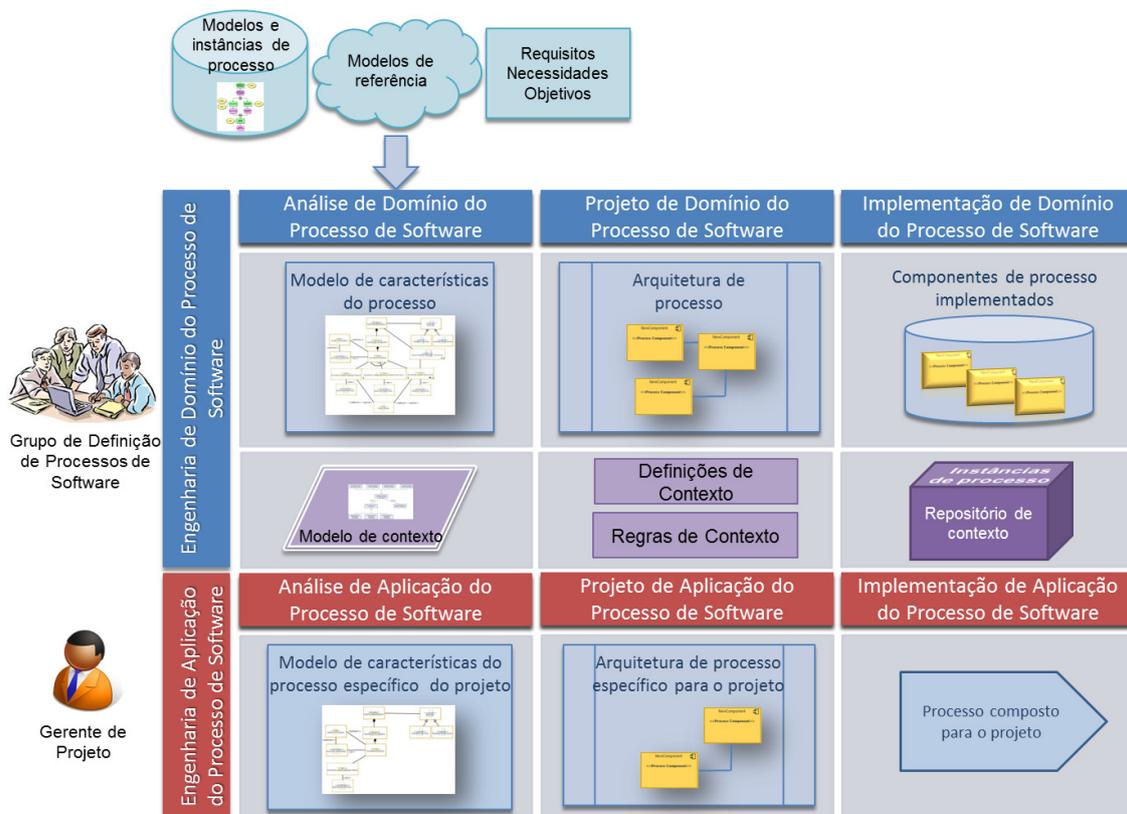


Figura 4.8 – Método da LPBC

4.4.1. Engenharia de Domínio de Processo (EDP)

A primeira fase (EDP) provê um conjunto de artefatos e infraestrutura para a derivação de múltiplos processos na próxima fase de EAP. Os principais objetivos da EDP são: (i) definir o escopo da LPS; (ii) definir as similaridades e variabilidades da LPS; (iii) definir e construir artefatos reutilizáveis que atendam a variabilidade desejada. Para alcançar estes objetivos, a EDP é composta pelas atividades de análise, projeto e implementação do domínio do processo

(SCHNIEDERS e PUHLMANN, 2006) (Figura 4.8). A representação do contexto faz parte da criação da LPS e, portanto, também acontece na EDP.

4.4.1.1. Análise de Domínio do Processo

A primeira atividade, análise de domínio do processo, busca identificar, coletar e organizar informações relevantes do domínio. O objetivo é gerar, a partir dos insumos fornecidos pelas estratégias de captura de conhecimento (apresentadas na Seção 4.2.2), o **modelo de características do processo**.

Este modelo define a variabilidade da LPS, introduzindo os pontos de variação e as variantes oferecidas em cada um deles. Também define as dependências e restrições que devem ser consideradas posteriormente durante a derivação de processos. As **regras de composição de características** são estabelecidas para definir estas restrições entre as características.

Em relação ao contexto, ainda nessa atividade deve ser criado o modelo de contexto contendo as dimensões e informações de contexto e as relações entre elas (conforme visto na Seção 4.3 e ilustrado pela Figura 4.7).

4.4.1.2. Projeto de Domínio do Processo

O projeto de domínio do processo é a atividade em que modelos de projeto são construídos, com base nos modelos de análise. Esta atividade compreende a criação de componentes, mapeamento entre os componentes e as características e geração da arquitetura de processos, a qual será reutilizada por diferentes processos derivados deste domínio.

A **arquitetura de processos** (Figura 4.8) descreve os componentes de processos reutilizáveis e as suas interfaces. A arquitetura de processos, representada através do modelo de componentes de processo, mostra os componentes, suas propriedades internas (opcionalidade e variabilidades) e interfaces (requeridas ou providas). Vale ressaltar que no trabalho de Barreto (2011), a arquitetura de processos é utilizada com semântica diferente, pois corresponde ao fluxo do processo.

Finalizando a atividade de projeto do domínio, são estabelecidas as **regras de composição de componentes**. Estas regras restringem ou obrigam a seleção de determinados componentes. As regras ajudam a garantir que sempre que um componente for escolhido para reutilização, os componentes relacionados também possam ser selecionados. Do mesmo modo, dois

componentes que sejam mutuamente exclusivos devem ser impedidos de serem utilizados conjuntamente.

Dando continuidade à representação dos elementos de contexto devem ser criadas as situações e regras de contexto (vistas na Seção 4.3). Estas regras de contexto vão estabelecer as relações entre o contexto e as características da LPS.

4.4.1.3. Implementação de Domínio do Processo

Na implementação de domínio do processo, a arquitetura de processos é efetivamente implementada no ambiente escolhido (Figura 4.8), para ser utilizada na composição dos processos. A LPS é, de fato, criada.

Esta atividade também envolve a implementação do repositório de contexto. O repositório de contexto servirá para armazenar os contextos associados à organização, projeto e equipe.

4.4.2. Engenharia de Aplicação do Processo (EAP)

Após criar a LPS na fase anterior, na EAP a organização passará a utilizá-la. Para cada novo projeto, ao invés de criar um processo manualmente, a organização usa a LPS original para a composição de múltiplos processos. Na composição do processo, são tomadas decisões que devem levar em consideração o contexto da organização, do projeto e da equipe (HENDERSON-SELLERS, 2002).

Os principais objetivos da EAP são: (i) alcançar um alto nível de reuso ao derivar um processo a partir da linha; (ii) explorar as similaridades e variabilidades do processo, modeladas no domínio, durante os recortes; (iii) documentar os artefatos da fase de aplicação e relacioná-los com os artefatos do domínio criados na fase anterior; (iv) tomar decisões sobre as variabilidades de acordo com as necessidades do projeto.

A EAP apresenta atividades análogas àquelas apresentadas na EDP. No entanto, uma vez que a EAP consiste na reutilização de artefatos gerados na EDP, são enfatizadas as particularidades de cada processo.

Ao final desta fase, o processo é recortado e fica pronto para ser executado, associado ao seu contexto. A partir daí, a GC vai verificar continuamente se, a partir dos novos contextos que forem aparecendo, as situações de contexto mudaram para alertar que a composição indicada para o

processo pode precisar ser revista.

Também é importante que a EAP forneça *feedback* para a EDP como forma de melhorar os artefatos e garantir que a infraestrutura provida se mantenha adequada para a composição de novos processos.

4.4.2.1. Análise de Aplicação do Processo

A atividade de análise de aplicação do processo inicia a EAP, ou seja, inicia o uso da LPS para a composição dos processos. O objetivo é obter um processo adequado ao projeto a partir da LPS criada. Neste momento, são aplicadas: as regras de composição de características; as regras de contexto para seleção de características a partir do contexto do projeto; e as verificações de consistência.

A tarefa de obtenção de um processo a partir da LPS é chamada de derivação ou recorte. Durante o recorte do modelo de características, as características que são mandatórias no domínio já são selecionadas. Cabe ao gerente de projeto selecionar as características opcionais desejadas. Ao final, o artefato gerado é o **modelo de características recortado** (Figura 4.8), ou seja, contendo apenas as características selecionadas para o processo em questão.

O modelo de características possui um alto nível de abstração, sendo utilizado como ponto de partida para o recorte dos novos processos. A partir do modelo de características, o recorte vai descendo gradualmente em diferentes níveis de abstração. Este recorte é influenciado pela variabilidade modelada no domínio e pela rastreabilidade estabelecida entre os diferentes artefatos. Isso significa que a seleção de uma característica influencia nas atividades posteriores, pois implica na inclusão desta e dos artefatos a ela relacionados, como os componentes de processos, que são automaticamente selecionados para o processo.

4.4.2.2. Projeto de Aplicação do Processo

Nesta atividade, a arquitetura criada é usada para instanciar a arquitetura correspondente ao processo atual. Isto significa que nesta atividade cabe ao gerente do projeto, levando em consideração características da organização, do projeto e da equipe, **recortar os componentes de processos** (Figura 4.8), selecionando aqueles que mais se adéquem ao contexto.

A partir do modelo de componentes recortado e das regras de composição existentes, o novo **processo é composto** (Figura 4.8). A ligação

entre os componentes de processo de software ocorre através de suas interfaces. Ao término do recorte dos artefatos, ajustes podem ser realizados. O gerente de projeto pode incorporar mudanças específicas para o processo em questão, incluindo ou excluindo alguns componentes de processo.

4.4.2.3. Implementação de Aplicação do Processo

Na última atividade da EAP, ou seja, na implementação de aplicação do processo, o **processo** composto é efetivamente **implementado** e está pronto para ser executado e adaptado em tempo de execução (Figura 4.8). Neste caso, pode-se optar, por exemplo, pela adoção de uma ferramenta de *workflow* ou BPMS (*Business Process Management System*) (ARAUJO e BORGES, 2001, MAGDALENO *et al.*, 2007b, PUNTAR *et al.*, 2009) para automatizar a execução do processo. Esta ferramenta pode ser, por exemplo, a Charon (MURTA, 2002) que oferece recursos para a modelagem, instanciação, simulação, execução, acompanhamento e evolução de processos de software.

É importante destacar que apesar da abordagem LPBC ser sistemática, o que pode causar a falsa impressão de rigor na definição dos processos, as características e os componentes de processos podem refletir práticas de qualquer um dos modelos de desenvolvimento vistos no Capítulo 3.

4.5. Composição e Otimização de Processos

Uma vez que a abordagem LPBC está em funcionamento, ou seja, já é capaz de sugerir processos para um projeto específico considerando o seu contexto, observa-se que ainda existem vários passos que dependem de decisões do gerente de projeto sobre variabilidades e opcionalidades. Assim, surge a necessidade de apoiar o gerente de projeto nas decisões sobre o recorte das características e componentes para a composição do processo de acordo com o contexto do projeto.

Apesar de ser uma das principais tarefas executadas pelo gerente do projeto, a composição do processo não é simples (TERNITE, 2009). Esta tarefa envolve o conhecimento de muitos aspectos da Engenharia de Software e requer a harmonização de muitos fatores do contexto da equipe, do projeto ou da organização (BARRETO, 2007). Compreender esses fatores e como eles poderão afetar o processo de software é um desafio para o gerente de projeto

(XU e RAMESH, 2008), principalmente para aqueles com menos experiência.

A experiência da área de Linhas de Produtos mostra que o recorte da linha não é uma tarefa fácil e pode chegar a ser cara e demorada (DEELSTRA *et al.*, 2005), especialmente no caso de linhas grandes ou múltiplas (MARQUES JUNIOR *et al.*, 2011). As mesmas dificuldades são esperadas para a composição de processos a partir da LPS, que pode se tornar uma tarefa complexa, demorada e propensa a erros quando realizada sem o devido apoio computacional (COSTA, 2010). Segundo Aleixo *et al.* (2010) e Barreto (2011), existe uma forte demanda por ferramentas que apoiem a composição de um processo customizado para o projeto.

No Projeto CDSOFT defende-se que é possível construir um apoio computacional para semi-automatizar a composição de um processo aderente ao contexto do projeto, inferindo e sugerindo quais componentes de processo são adequados ao projeto, possivelmente diminuindo o esforço necessário para sua execução e melhorando os resultados obtidos.

Este apoio deve fornecer ao gerente um melhor embasamento para a tomada de decisões durante a composição de processos, uma vez que possibilita que seja considerado um universo maior de alternativas. O apoio sugerido diz respeito a propor ao gerente de projeto opções de processo que satisfaçam as restrições e que otimizem aspectos de interesse da organização. Porém, a decisão final sobre o processo que será efetivamente adotado continua sendo do gerente.

A seleção da melhor combinação de componentes de processo, atendendo às restrições impostas pelo contexto do projeto, é um problema de otimização. Assim, o ferramental de apoio do Projeto CDSOFT foi construído utilizando como base métodos de otimização que têm como finalidade buscar soluções que atendam um conjunto de restrições impostas, para maximizar ou minimizar um determinado aspecto de interesse (PAPADIMITRIOU e STEIGLITZ, 1998).

Dependendo do objetivo, diferentes aspectos podem ser otimizados. Pelos motivos já expostos nos capítulos anteriores, a colaboração é o aspecto que se pretende otimizar no escopo desta pesquisa de doutorado, conforme destacado na próxima seção.

4.6. COMPOOTIM no Projeto CDSOft

O objetivo principal da COMPOOTIM é planejar, otimizar e acompanhar a colaboração na composição de processos de software. A COMPOOTIM visa oferecer uma solução voltada para a otimização da colaboração na composição de processos de software.

No Projeto CDSOft, a COMPOOTIM funcionou como uma instância completa, que construiu e exercitou todas as etapas da visão de pesquisa do projeto, com foco no aspecto da colaboração (Figura 4.9). Assim, a COMPOOTIM contribuiu para os diferentes objetivos do projeto, ao mesmo tempo em que demonstrou a sua viabilidade ao conseguir construir um ferramental de apoio com mecanismos para a composição, otimização e visualização do processo.

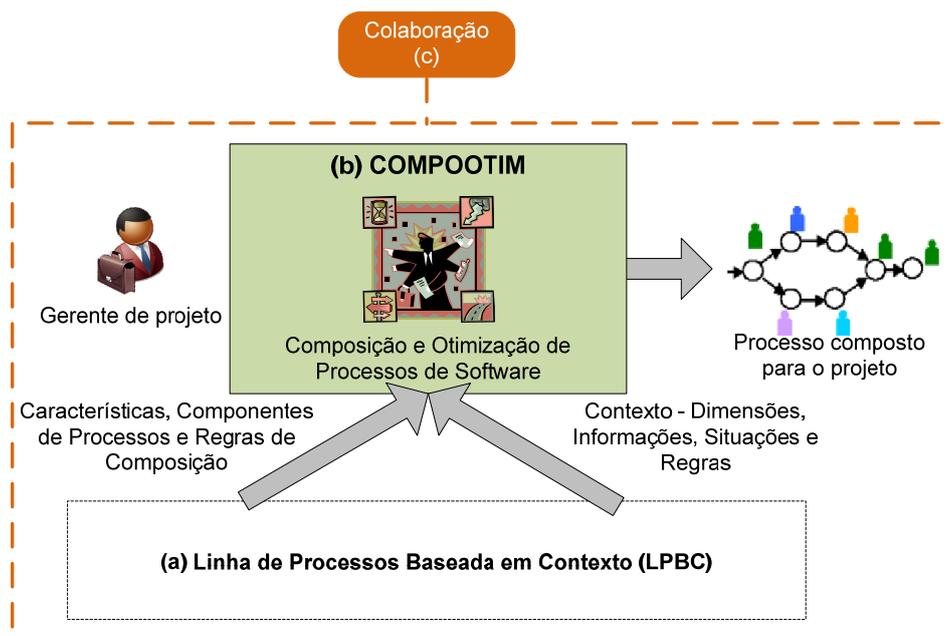


Figura 4.9 – COMPOOTIM no contexto do Projeto CDSOft

Ao delinear o arcabouço geral do projeto e percorrer todas as soluções previstas no Projeto CDSOft com o intuito de planejar, otimizar e acompanhar a colaboração, esta tese abriu espaço para outros trabalhos de pesquisa (de mestrado e doutorado), conforme ilustrado pela Figura 4.10. As ramificações apresentadas mostram que as contribuições desta tese de doutorado não se restringem as soluções construídas na COMPOOTIM, pois o envolvimento da autora como pesquisadora ativa em todos estes trabalhos agregou também outros resultados para o Projeto CDSOft como um todo.

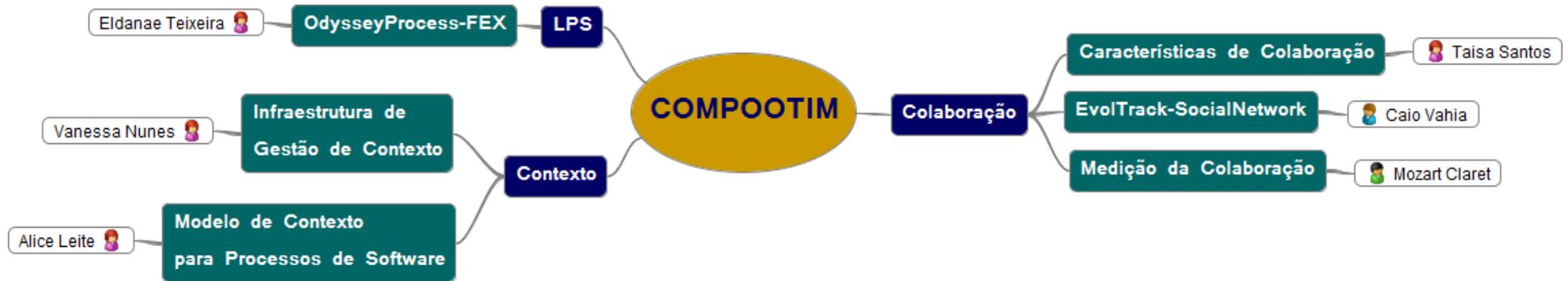


Figura 4.10 – Mapa de trabalhos de pesquisa ramificados a partir da COMPOOTIM

Os trabalhos relacionados à LPS e Contexto já foram resumidos nas seções anteriores. Em relação à colaboração, as características de colaboração em redes sociais foram desenvolvidas por Santos (2010), com a participação da autora desta tese como coorientadora informal, e são utilizadas nesta tese como um dos instrumentos para o acompanhamento da colaboração. A autora também participou como coorientadora no projeto de iniciação científica (VAHIA *et al.*, 2011) que implementou a ferramenta EvolTrack-SocialNetwork concebida para o acompanhamento das redes sociais nos projetos de software. Por fim, um trabalho mais recente e ainda em andamento deverá partir das propostas de medição da colaboração desta tese para dar continuidade às investigações sobre o tema (CLARET, 2013).

4.7. Considerações Finais

Este capítulo apresentou o Projeto CDSOFT, descrevendo a visão geral da solução proposta para a composição de processos de software. Ao longo dos três anos de realização do projeto, ele funcionou como celeiro para 3 teses de doutorado (sendo esta a primeira delas concluída), 11 dissertações de mestrado (sendo 10 já concluídas) e 2 projetos de iniciação científica também finalizados. Neste período, o projeto também rendeu mais de 15 publicações, sendo alguns dos resultados obtidos compartilhados com a comunidade científica e divulgados em conferências e revistas nacionais e internacionais.

Atualmente, o grupo de pesquisa está concentrado no detalhamento e construção da infraestrutura de GC e na atividade de projeto de domínio da LPBC. Estas iniciativas são capitaneadas por outras duas teses de doutorado (NUNES, 2011b) em andamento no projeto. Os próximos passos do projeto estão direcionados para a integração entre os ambientes construídos (Odyssey, a COMPOOTIM e a infraestrutura de GC) como parte da solução proposta para a composição de processos. Além disso, também é intenção que o ambiente integrado construído evolua para se tornar um ambiente de apoio à decisão durante a composição de processos de software.

Uma vez entendido o papel desta tese no contexto do Projeto CDSOFT, o próximo capítulo trata de detalhar as soluções para planejar, compor, otimizar e acompanhar a colaboração.

5. COMPOOTIM

Este capítulo detalha a proposta da COMPOOTIM para o planejamento, composição, otimização e acompanhamento da colaboração em processos de software. Em particular, são apresentadas as soluções desenvolvidas ou aplicadas para apoiar cada uma destas etapas.

Para lidar com a colaboração como um aspecto da diversidade de projetos e processos de desenvolvimento de software, foi concebida a COMPOOTIM. A visão construída na COMPOOTIM prevê as etapas de planejamento, composição, otimização e acompanhamento da colaboração, conforme apresentado na Figura 5.1. Cada uma destas etapas foi criada para resolver alguns desafios da introdução da colaboração nos processos de software. Para cada uma destas etapas, foram desenvolvidas ou aplicadas soluções, que serão descritas nas seções seguintes.

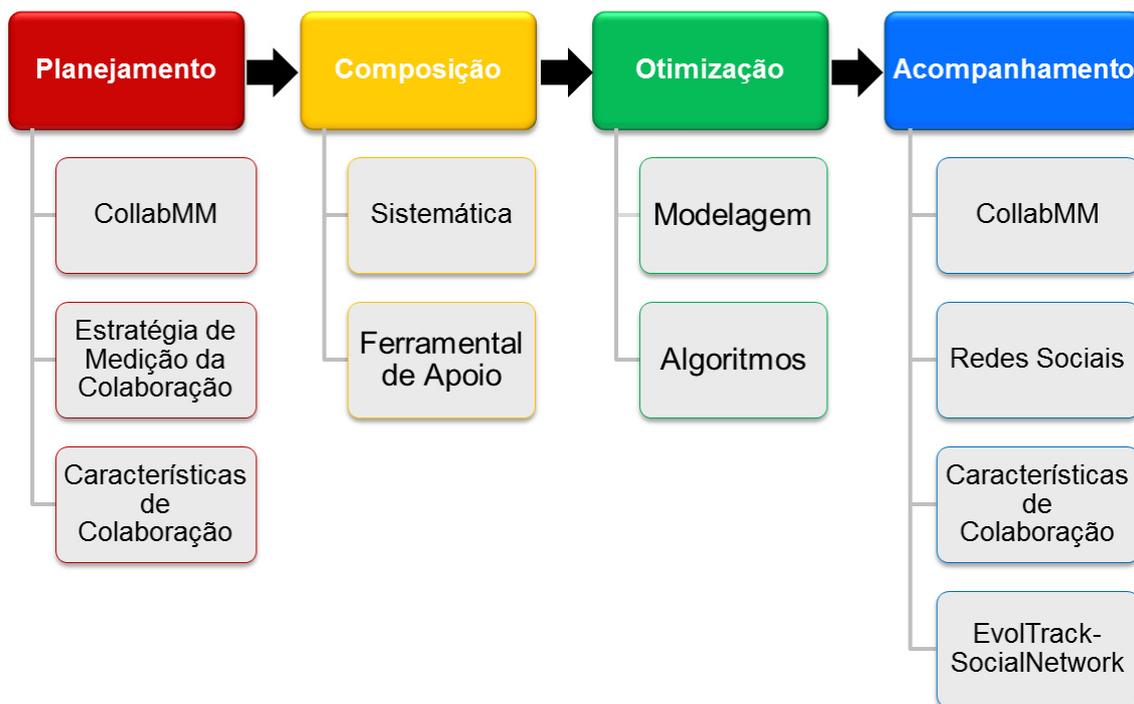


Figura 5.1 – Visão Geral das Etapas e Soluções na COMPOOTIM

Na etapa de **planejamento da colaboração**, o primeiro desafio era: “*Como definir os objetivos da colaboração?*”. A resposta encontrada foi o uso do Modelo de Maturidade em Colaboração (CollabMM) como uma referência, definindo os níveis de colaboração em processos e resumindo as principais características de cada nível (Seção 5.1.1). Além disso, era preciso definir: “*Como calcular a colaboração do processo?*”. Neste caso, foi proposta uma

estratégia de medição da colaboração nos componentes individuais e no processo como um todo (Seção 5.1.2).

Na etapa de **composição da colaboração**, para vencer o desafio de “*Como guiar a composição do processo?*”, foi desenvolvida uma sistemática para criação e uso de linha de processos (Seção 5.2.1). O desafio de “*Como automatizar a composição de processos?*” foi resolvido com a construção do ferramental de apoio com mecanismos de composição, otimização e visualização (Seção 5.2.2).

Na etapa de **otimização da colaboração**, era preciso analisar a seguinte questão: “*Como projetar a colaboração como um problema de otimização?*”. Este desafio foi superado com a modelagem formal do problema (Seção 5.3.2). O passo seguinte era definir “*Como implementar a otimização da colaboração em processos de software?*”. Para isso, foram utilizados os algoritmos de Força Bruta, *Random Search* e *Hill Climbing* (Seção 5.3.3).

Por fim, na etapa de **acompanhamento da colaboração**, o CollabMM foi novamente utilizado na questão de “*Como monitorar o nível de colaboração no processo?*”. Além disso, também era necessário saber “*Como verificar se a colaboração está de fato acontecendo conforme o planejado?*”. A solução para este desafio foi o uso das redes sociais, que combinadas com o CollabMM, deram origem às características de colaboração (Seção 5.4.1). Por fim, observou-se que este acompanhamento feito de forma manual não seria viável. Então, era necessário definir “*Como apoiar computacionalmente a visualização e análise das redes sociais?*”. Este último desafio deu origem à ferramenta EvolTrack-SocialNetwork (Seção 5.4.2).

O restante deste capítulo está organizado da seguinte forma: a Seção 5.1 detalha os instrumentos de planejamento da colaboração; a Seção 5.2 é dedicada aos instrumentos de composição; a Seção 5.3 está voltada para os instrumentos de otimização; a Seção 5.4 aborda os instrumentos de acompanhamento da colaboração; a Seção 5.5 foca no apoio computacional às soluções propostas; por fim, a Seção 5.6 tece considerações finais ao capítulo.

5.1. Planejamento da Colaboração

O planejamento da colaboração deve permitir às organizações de software

estabelecer objetivos e definir ações para atingir níveis mais altos de colaboração. A composição inicial do processo é feita pelo gerente como parte das atividades de planejamento do projeto. Portanto, neste momento, deve ser planejado também o nível de colaboração necessário ou desejado para o projeto. Além disso, é conveniente que a composição do processo seja revista durante a execução do projeto de desenvolvimento de software devido às mudanças que vão acontecendo em seu contexto. Neste momento, é preciso avaliar se o nível planejado de colaboração ainda se mantém o mesmo e, se for o caso, rever o planejamento feito anteriormente.

5.1.1. Modelo de Maturidade em Colaboração (CollabMM)

O Modelo de Maturidade em Colaboração (*Collaboration Maturity Model - CollabMM*) foi proposto em um trabalho prévio (MAGDALENO, 2006, MAGDALENO *et al.*, 2007a, 2008a, 2008b, 2009a) e é usado na COMPOOTIM para planejar o nível de colaboração desejado ou necessário para o processo que será composto para o projeto. Ele funciona como uma referência, definindo os níveis de colaboração em processos e resumindo as principais características de cada nível.

Este modelo foi desenvolvido analogamente a outros modelos de maturidade existentes em outros domínios (CHRISISS *et al.*, 2006, EHMS e LANGEN, 2002, FISHER, 2004, PAULK *et al.*, 1994, ROSEMAN *et al.*, 2004). A base conceitual do modelo foi extraída da literatura de CSCW (ALTMANN e POMBERGER, 1999, ELLIS *et al.*, 1991, GEROSA *et al.*, 2006, GREIF, 1988, KHOSHAFIAN e BUCKIEWICZ, 1995, MISTRİK *et al.*, 2010, PIMENTEL e FUKS, 2011), cuja pesquisa organizou o corpo de conhecimento sobre colaboração em quatro aspectos de apoio a colaboração (comunicação, coordenação, memória e percepção), detalhados no Capítulo 2.

O CollabMM funciona como um *framework* que organiza as principais práticas, tanto as existentes na literatura quanto as adotadas nas organizações, que podem ser aplicadas aos processos para garantir/estimular a colaboração. Este modelo foi escolhido para ser utilizado na COMPOOTIM por diversos motivos: trata-se de um modelo simples (unidimensional), que aborda a colaboração através dos processos e dentro de uma organização, conforme desejado no contexto deste trabalho; já está estruturado em níveis e práticas

de colaboração; é de conhecimento da pesquisadora; está inserido no contexto do grupo de pesquisa; e já foi aplicado em diferentes ambientes reais (MAGDALENO *et al.*, 2009a, 2012b).

5.1.1.1. Níveis de Maturidade em Colaboração

O CollabMM descreve um caminho de evolução progressiva através de quatro níveis de maturidade em colaboração: *casual*, *planejado*, *perceptivo* e *reflexivo* (Figura 5.2). A estruturação em níveis de maturidade é uma forma de ajudar as organizações a priorizarem as ações para estimular a colaboração. Para se alcançar um determinado nível, é necessário satisfazer este nível e os seus níveis inferiores.

Cada um dos níveis de colaboração, exceto o casual, engloba um conjunto de objetivos e práticas de colaboração que auxiliam a sua aplicação. Ao todo, o modelo descreve 12 (doze) práticas de colaboração distribuídas em três níveis de maturidade: planejado (4), perceptivo (4) e reflexivo (4), conforme pode ser visto na Figura 5.2. Não existem práticas para o nível casual, uma vez que neste nível não se espera que a colaboração esteja formalmente estabelecida na organização.

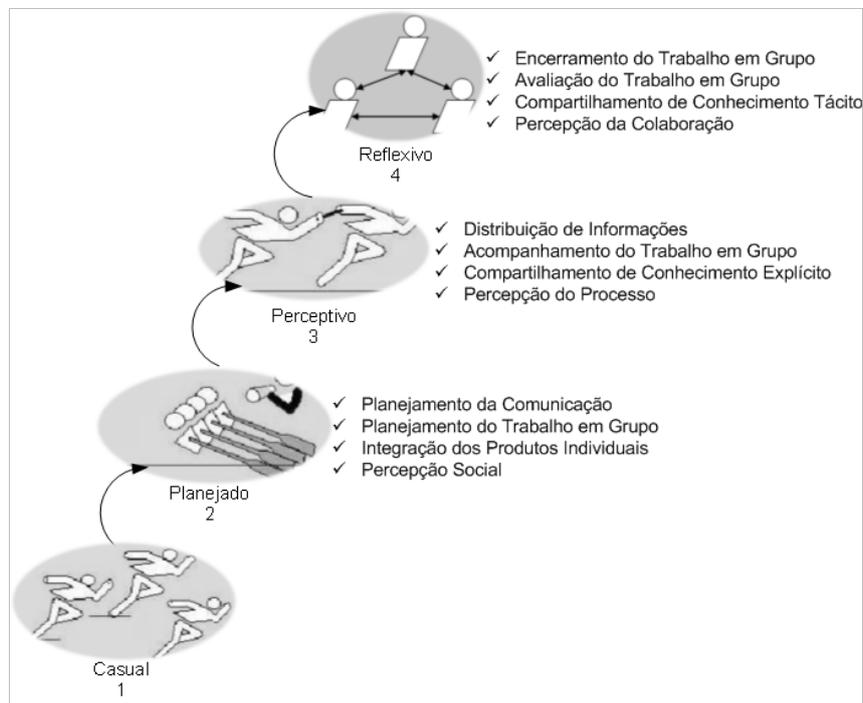


Figura 5.2 – Níveis e Práticas do CollabMM

a) Nível 1 – Casual

No nível casual, a colaboração ainda não está explícita nos processos da organização. Contudo, as organizações no nível 1 não se caracterizam pela

total ausência de colaboração. A colaboração acontece, mas como uma prática isolada e dependente do relacionamento ou da afinidade existentes entre as pessoas.

Assim, a colaboração ainda é resultado do esforço individual e acontece de forma casual. Nestas situações, os indivíduos gastam um grande esforço em sua coordenação, pois trabalham independentemente uns dos outros. Os aspectos de comunicação, coordenação, memória e percepção estão presentes, mas são tratados de maneira *ad-hoc*, ou seja, sem sistematização. Neste caso, é difícil prever os resultados conjuntos ou aprender e reproduzir as experiências.

b) Nível 2 – Planejado

A coordenação é um aspecto forte deste nível e acontece de forma centralizada por um membro que assume o papel de líder e age de acordo com a sua experiência. O líder estimula o comprometimento e a moral do grupo, encorajando os membros e comunicando claramente os papéis e responsabilidades. Este líder é o responsável pelo planejamento do trabalho e distribuição de tarefas entre os participantes do grupo. Cabe ao líder balancear a carga de trabalho dos participantes do grupo, ao mesmo tempo em que procura aproveitar as habilidades e talentos específicos de cada um.

Os indivíduos já são capazes de: se perceberem como parte integrante do grupo, compreender a formação deste grupo e conhecer os seus parceiros de trabalho. Com as informações obtidas sobre os participantes, o indivíduo já consegue estabelecer as conexões sociais, ajudando a criar solidariedade, entendimento, confiança, respeito e compromisso no âmbito de cada grupo do qual participe (ARAUJO, 2000, SANTORO, 2001).

Também é importante garantir que os produtos do grupo não sejam apenas resultados de esforços individuais isolados e nem uma simples composição de trabalhos distintos. Mesmo que exista a divisão de tarefas entre os colaboradores visando gerar maior rapidez na execução de uma atividade, os artefatos devem ser integrados, de forma a se complementarem como parte de um todo consistente, entendido por todo o grupo (SANTORO, 2001).

c) Nível 3 – Perceptivo

No nível perceptivo, os membros do grupo conhecem as suas

responsabilidades e sabem quais atividades executar de forma que o grupo consiga alcançar os seus objetivos. Assim, não é mais necessária uma coordenação centralizada do líder. O esforço passa a ser distribuído. Os membros do grupo trabalham de forma auto-organizada e simultânea para atingir os objetivos do grupo. Por outro lado, é necessário que se garanta ao grupo os recursos necessários para acessar as informações e entender a dependência e a articulação das suas atividades.

Os membros do grupo compreendem o processo do trabalho que irão realizar, entendem seus objetivos, estão conscientes dos passos necessários para alcançar estes objetivos e possuem os conhecimentos necessários para executar as tarefas.

O conhecimento explícito (NONAKA e TAKEUCHI, 1995) é compartilhado, na forma de artefatos, entre os membros do grupo. Para apoiar este compartilhamento, são claramente definidos os artefatos manipulados e o repositório para armazenamento desta documentação.

d) Nível 4 – Reflexivo

As organizações que atingem o nível reflexivo de maturidade nos seus processos são aquelas que percebem o valor do conhecimento que está sendo gerado no trabalho dos grupos e se preocupam em geri-lo e disseminá-lo dentro da própria organização. A colaboração é institucionalizada na empresa, passando a ser vista como fonte de conhecimento.

Os resultados do trabalho e o conhecimento gerado no processo são analisados, compartilhados e disseminados na organização. Neste nível, o conhecimento tácito (NONAKA e TAKEUCHI, 1995) também passa a ser compartilhado, na forma de ideias, opiniões e experiências, entre os membros do grupo.

Os participantes do processo são capazes de se auto-direcionarem para executar as tarefas e determinarem o que precisam para realizar o trabalho. Os participantes devem perceber com clareza como a colaboração acontece na execução do processo, ou seja, como suas atividades interagem entre si, de forma que cada um possa pautar suas próprias contribuições.

A Tabela 5.1 resume as práticas de colaboração do CollabMM, classificando-as de acordo com os aspectos de apoio a grupos e os níveis de

maturidade a que elas se referem.

Tabela 5.1 – Classificação das Práticas de Colaboração de acordo com os Aspectos de Apoio à Colaboração e os Níveis de Maturidade

	Comunicação	Coordenação	Memória	Percepção
Reflexivo (4)	Encerramento	Avaliação	Compartilhamento de Conhecimento Tácito	Percepção da Colaboração
Perceptivo (3)	Distribuição das Informações	Acompanhamento do Trabalho	Compartilhamento de Conhecimento Explícito	Percepção do Processo
Planejado (2)	Planejamento da Comunicação	Planejamento do Trabalho em Grupo	Integração dos Produtos Individuais	Percepção Social

5.1.2. Estratégia de Medição da Colaboração

A medição é o “processo através do qual valores (símbolos ou números) são atribuídos a atributos de entidades do mundo real, de modo a descrevê-los de acordo com regras claramente definidas” (ROBERTS, 2009). Ou, colocando de outra forma, a medição corresponde a um “conjunto de operações que tem como objetivo determinar o valor de uma medida” (ISO/IEC, 2007).

O principal objetivo da medição em gerenciamento de projetos é apoiar as necessidades de informações gerenciais (CHRISSIS *et al.*, 2006). Neste sentido, pode-se considerar que a medição favorece a tomada de decisão, pois esta pode ser feita com base em informações reais do projeto e históricas da própria organização.

Geralmente, as organizações usam a medição (através de indicadores e métricas) no dia-a-dia para medir e acompanhar os seus processos. Do mesmo modo, também é necessário medir objetivamente a colaboração. No entanto, este é um grande desafio e existem poucas propostas concretas que definam como medir a colaboração (THOMSON *et al.*, 2009).

Apesar de existirem alguns trabalhos sobre medição da colaboração (ARAUJO *et al.*, 2002, DAVID, 2004, PEREIRA e SOARES, 2007), não foram encontradas soluções específicas para a definição de métricas e indicadores de desempenho da colaboração, durante e/ou após a execução de processos.

Neste trabalho, o CollabMM foi usado como referência para definição da estratégia de medição da colaboração, pois estrutura as práticas de colaboração a partir das quais se propõe quantificar a colaboração. A

estratégia de medição da colaboração é parte do planejamento da colaboração, pois define como é calculado o potencial de colaboração de cada componente de processo e a colaboração do processo como um todo.

5.1.2.1. Potencial de Colaboração dos Componentes de Processo

Seguindo a abordagem de Linha de Processos Baseada em Contexto (LPBC), apresentada no capítulo anterior, ao criar a LPS, são definidos os componentes de processo que compõem esta linha. A estrutura de definição de um componente de processo prevê que seja estabelecido o potencial de colaboração (PC) de cada componente. Esta seção apresenta a proposta para a definição do PC de um componente de processo.

O primeiro passo é avaliar o componente de processo, vendo se ele contribui ou não para alguma(s) das práticas de colaboração definidas no CollabMM (listadas na Tabela 5.2) e se contribui para qual(is) dela(s). Cada uma das práticas tem um peso (p) associado. Este peso foi definido com base nos níveis de maturidade em colaboração do CollabMM.

Tabela 5.2 – Aspectos de Apoio e Práticas de Colaboração

Aspectos de Apoio	Práticas de Colaboração		
	Nível 2	Nível 3	Nível 4
Comunicação	Planejamento da Comunicação (p=2)	Distribuição das Informações (p=3)	Encerramento (p=4)
Coordenação	Planejamento do Trabalho em Grupo (p=2)	Acompanhamento do Trabalho (p=3)	Avaliação (p=4)
Memória	Integração dos Produtos Individuais (p=2)	Compartilhamento de Conhecimento Explícito (p=3)	Compartilhamento de Conhecimento Tácito (p=4)
Percepção	Percepção Social (p=2)	Percepção do Processo (p=3)	Percepção da Colaboração (p=4)

Para ilustrar o entendimento desta estratégia de medição de colaboração suponha o componente de processo “Realizar Reunião Diária de Acompanhamento” apresentado na Tabela 5.3 que coloca o papel, descrição, característica associada, opcionalidade, variabilidade, interfaces requeridas e interfaces providas.

Durante uma reunião diária, cada membro explica o que realizou desde a última reunião diária, o que fará até a próxima reunião e quais são os

obstáculos que está encontrando para realizar o trabalho. Desta forma, o *status* das tarefas é atualizado.

Tabela 5.3 – Componente de Processo Realizar Reunião Diária de Acompanhamento

CP5 – Realizar Reunião Diária de Acompanhamento	
Papel:	Time do Projeto
Descrição:	Tem como objetivo melhorar a comunicação, identificar e remover impedimentos e riscos para o desenvolvimento, ressaltar e promover a tomada rápida de decisões e aumentar o nível de conhecimento de todos acerca do projeto. É fortemente recomendado que os membros do time se encontrem diariamente para uma reunião de aproximadamente 15 minutos.
Característica:	Acompanhar Projeto Ágil
Opcionalidade:	Mandatário
Variabilidade:	Invariante
Interfaces Requeridas:	<i>Backlog</i> do Produto Visão do Projeto
Interfaces Providas:	Gráfico <i>Burndown</i> da Sprint Quadro de Acompanhamento da Sprint

Aplicando a estratégia de medição da colaboração, este componente foi avaliado conforme descrito na Tabela 5.4. Nesta avaliação, este componente foi considerado como estando relacionado com as seguintes práticas:

Tabela 5.4 – Exemplo de Avaliação do Potencial de Colaboração do Componente de Processo

Aspectos de Apoio	Práticas de Colaboração		
	Nível 2	Nível 3	Nível 4
Comunicação	Planejamento da Comunicação ()	Distribuição das Informações (X)	Encerramento ()
Coordenação	Planejamento do Trabalho em Grupo ()	Acompanhamento do Trabalho (X)	Avaliação ()
Memória	Integração dos Produtos Individuais ()	Compartilhamento de Conhecimento Explícito ()	Compartilhamento de Conhecimento Tácito (X)
Percepção	Percepção Social (X)	Percepção do Processo (X)	Percepção da Colaboração ()
PC	3.00		

- **Distribuição de Informações** – ao apresentar um resumo do andamento do trabalho de cada um, as informações fluem entre os membros da equipe de desenvolvimento e a reunião serve como uma oportunidade para os

membros do grupo interagirem e compartilhar os seus sucessos e problemas;

- **Acompanhamento de Trabalho** – contribui para a visibilidade do progresso do trabalho do grupo;
- **Compartilhamento de Conhecimento Tácito** – a reunião funciona também como um canal que permite a socialização entre os membros do grupo para que através do diálogo e do debate, eles possam compartilhar experiências, ideias, fatos, ou pontos de vista;
- **Percepção Social** – a reunião diária ajuda a garantir que os participantes compreendam a formação dos grupos dos quais fazem parte, os seus parceiros de trabalho e estabeleçam as conexões sociais e proximidade física entre eles;
- **Percepção do Processo** – a reunião também facilita que os participantes tenham consciência da estruturação do trabalho, ou seja, dos passos a serem dados para o cumprimento dos objetivos, e do papel de cada um dentro do processo.

Após a associação do componente de processo com as práticas de colaboração, o PC do componente é calculado aplicando uma mediana ponderada dos pesos das práticas do CollabMM observadas no componente, o que resulta em um PC = 3.00, conforme apresentado na Tabela 5.4.

Vale ressaltar que uma limitação desta estratégia de medição é que ela não trabalha com PC negativo para os componentes de processo, mas na prática é possível imaginar que um determinado componente dificulte a colaboração e que ele poderia sim ser avaliado com um PC negativo.

Além disso, devido ao caráter subjetivo inerente à colaboração, também não é possível afirmar que esta estratégia de medição seja independente do avaliador, pois pode haver interpretações diferentes sobre para quais práticas de colaboração, um componente de processo contribui.

5.1.2.2. Nível de Colaboração do Processo

Para o planejamento da colaboração, é preciso que seja definido como deve ser calculado o nível de colaboração do processo como um todo dada a combinação do potencial de colaboração dos seus componentes de processo. Assim, foram disponibilizadas quatro estratégias alternativas, cuja escolha

poderá ser feita pelo gerente de projeto como parte do planejamento da colaboração durante a composição de processos:

- **Estratégia Otimista** – esta estratégia assume que o nível de colaboração do processo corresponde ao maior PC entre os componentes que compõem o processo. A ideia nesta estratégia é que se o processo composto possui pelo menos um componente com PC mais alto, então todo o processo será capaz de atingir um nível mais elevado de colaboração;
- **Estratégia Pessimista** – esta estratégia assume que o nível de colaboração do processo corresponde ao menor PC entre os componentes que compõem o processo. A ideia é que se o processo composto possui pelo menos um componente com PC mais baixo, então o processo não será capaz de atingir um nível mais elevado de colaboração, pois estará limitado pelos componentes de PC inferior;
- **Estratégia Mediana** – esta estratégia assume que o nível de colaboração do processo corresponde à mediana do PC dos componentes que compõem o processo. Nesta estratégia, o nível de colaboração do processo pode ser calculado considerando-se o PC de todos os componentes e tentando-se estimar um nível de colaboração médio;
- **Estratégia Acumulativa** – esta estratégia assume que o nível de colaboração do processo corresponde ao somatório do PC dos componentes que compõem o processo. Esta estratégia assume que a cada componente adicionado, o nível de colaboração do processo cresce de acordo com o respectivo PC. Porém, esta é uma medida não normalizada e não vai permitir comparar processos onde o número de componentes varia.

Considerando como exemplo um processo que inclua os componentes de processo apresentados na Tabela 5.5.

Tabela 5.5 – Exemplos de Componentes de Processo e seus respectivos PCs

Componente de Processo	Potencial de Colaboração (PC)
CP6 – Constituir Equipe para Execução do Projeto	2.00
CP7 – Elaborar Cronograma do Projeto	2.25
CP11 – Elaborar Plano de Comunicação	2.00
CP19 – Distribuir Informações do Projeto	3.00
CP20 – Acompanhar Desempenho do Projeto	3.75

Componente de Processo	Potencial de Colaboração (PC)
CP35 – Realizar Reunião Diária	2.00
CP40 – Detalhar Casos de Uso	3.00
CP61 – Executar Testes	4.00
CP26 – Registrar Lições Aprendidas	4.00
CP29 – Encerrar Projeto	4.00

Seguindo cada uma das estratégias apresentadas, o nível de colaboração deste processo teria resultados diferentes, conforme apresentado na Tabela 5.6. Na estratégia otimista, o nível de colaboração do processo seria 4.00, conforme o maior PC que é do CP26, CP29 e CP61. Na estratégia pessimista, o nível de colaboração do processo seria 2.00, conforme o menor PC que corresponde ao CP6, CP11 e CP35. Já na estratégia mediana, o nível de colaboração é calculado pela mediana dos PCs dos componentes, resultando no valor 3.00. Por fim, na estratégia acumulativa, o nível de colaboração do processo é apurado através do somatório dos PCs, o que corresponde ao valor 30.00.

Tabela 5.6 – Cálculo do nível de colaboração do processo em cada estratégia

Estratégia	Rationale	Nível de Colaboração do Processo
Otimista	Componente com maior PC	4.00
Pessimista	Componente com menor PC	2.00
Mediana	Mediana	3.00
Acumulativa	Somatório	30.00

5.2. Composição da Colaboração

A composição deve ser capaz de sugerir um processo específico para o projeto, a partir de componentes de processo, de acordo com o contexto. A composição de processos pode ser uma tarefa complexa, demorada e propensa a erros quando realizada sem o devido apoio (COSTA, 2010). Então, para apoiar o gerente de projeto na composição de um processo com colaboração, foi construída na COMPOOTIM tanto uma sistemática quanto um ferramental que auxiliem o gerente de projeto a conduzir esta composição. A sistemática segue a visão geral sugerida pela abordagem LPBC, apresentada

no Capítulo 4.

5.2.1. Sistemática

Esta seção apresenta a sistemática de composição de processos, incluindo as etapas de criação e uso da LPS. Todas as etapas foram representadas através de um diagrama de atividades da UML (*Unified Modeling Language*).

5.2.1.1. Criação da Linha de Processos Baseada em Contexto

Seguindo a sistemática da COMPOOTIM, a primeira etapa é a criação da LPS, na fase de Engenharia de Domínio de Processos. Um exemplo de criação da LPS, usando diferentes modelos de desenvolvimento de software pode ser encontrado em (MAGDALENO *et al.*, 2011b).

a) Análise de Domínio de Processos

Os passos e artefatos resultantes da fase de Análise de Domínio de Processos são detalhados no diagrama apresentado na Figura 5.3.

Seguindo esses passos, deve ser definida a LPS, considerando os objetivos, necessidades e requisitos da organização. Na COMPOOTIM, a LPS é definida seguindo a estrutura apresentada na Tabela 5.7. As entradas disponíveis dizem respeito aos artefatos que podem ser gerados em processos anteriores e estarão disponíveis para a LPS que está sendo definida. Os resultados esperados se referem aquilo que o processo deve ser capaz de entregar de resultado ao ser executado para qualquer projeto a partir da LPS.

Tabela 5.7 – Estrutura para definição de LPS

Campo	Descrição
Nome:	<Nome da LPS>
Descrição:	<Descrição da LPS >
Entradas Disponíveis:	<Artefatos disponíveis como entrada da LPS>
Resultados Esperados:	<Artefatos esperados como saída da LPS>

Em seguida, devem ser criadas as características de processo, observando as suas opcionalidades e variabilidades. O próximo passo é definir as regras de composição entre essas características. As regras criadas são validadas de acordo com as regras de boa formação definidas na *OdysseyProcess-FEX* (TEIXEIRA, 2011). Por último, deve ser iniciado o modelo de contexto com as dimensões e informações de contexto (segundo a estrutura da UbiFEX vista no Capítulo 4).

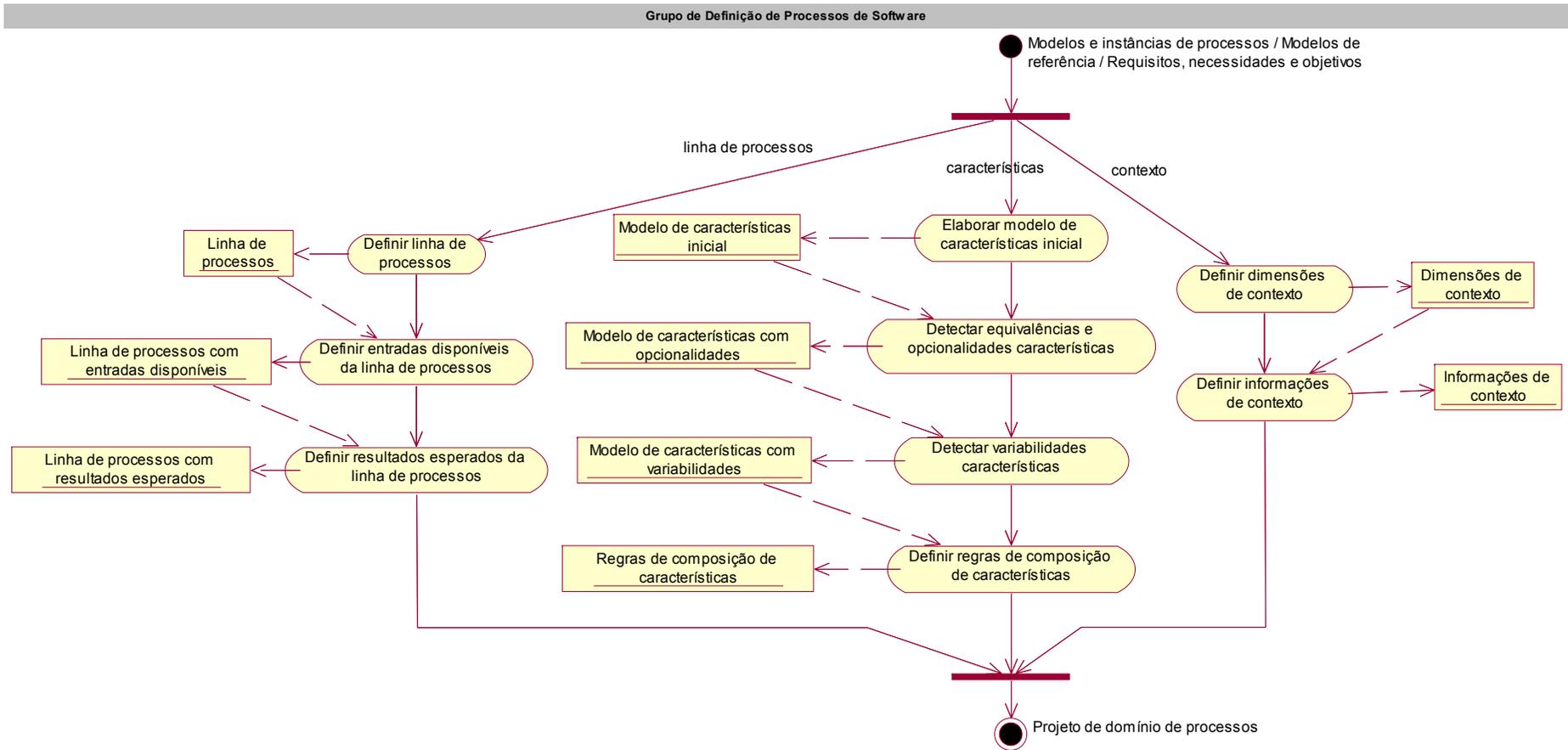


Figura 5.3 – Passos e artefatos da fase de Análise de domínio de processos

b) Projeto de Domínio de Processos

Os passos e artefatos resultantes desta fase são detalhados no diagrama apresentado na Figura 5.4. Na etapa Projeto de Domínio de Processos, deve ser criada a arquitetura de componentes de processo. Os componentes são associados às características de processos. Em seguida, devem ser definidas as opcionalidades e variabilidades dos componentes. Ainda para os componentes, deve ser definido o potencial de colaboração de cada um deles, de acordo com a estratégia para medição da colaboração apresentada na Seção 5.1.2.

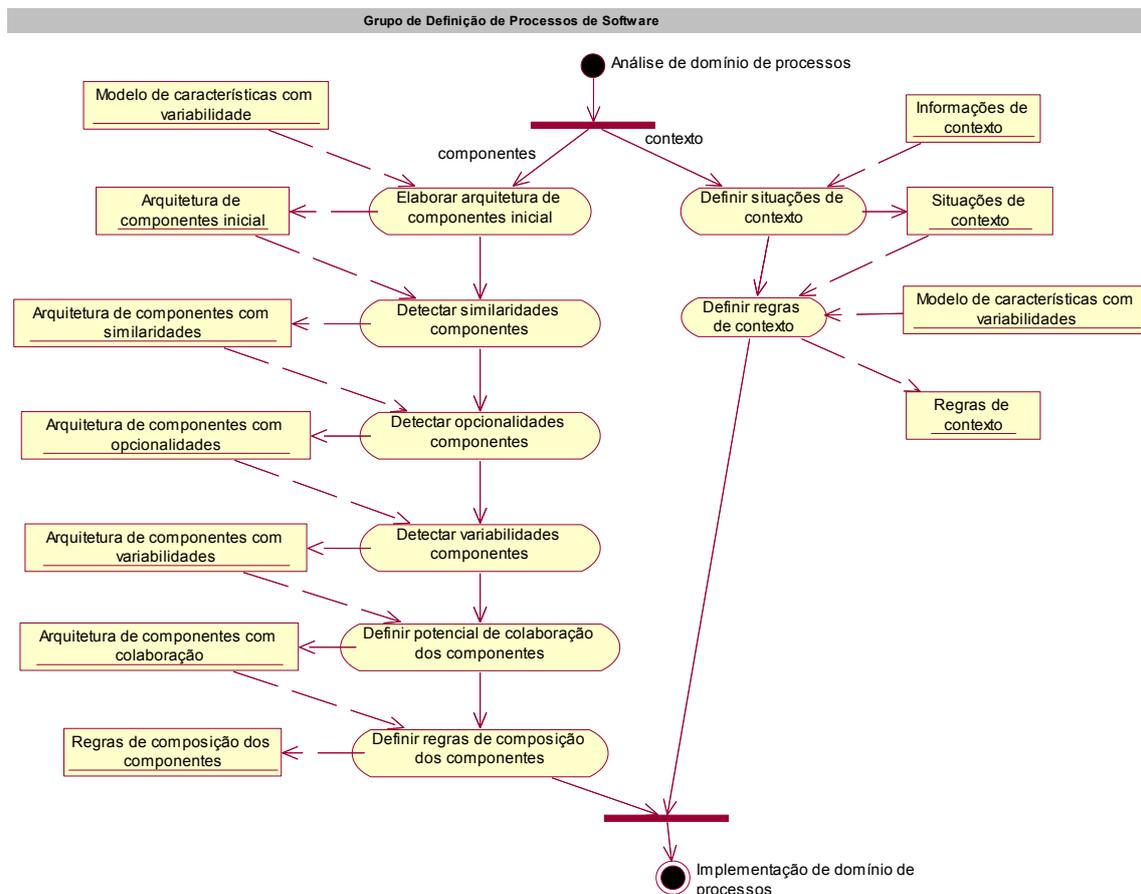


Figura 5.4 – Passos e artefatos da fase de Projeto de domínio de processos

Em seguida, são estabelecidas as regras de composição de componentes. Por fim, o modelo de contexto é complementado com as situações e regras de contexto.

5.2.1.2. Uso da Linha de Processos Baseada em Contexto

Seguindo a sistemática da COMPOOTIM, a segunda etapa é o uso da LPS para a composição de processos, na fase de Engenharia de Aplicação.

a) Análise de Aplicação de Processos

Os passos e artefatos resultantes desta fase são detalhados no diagrama apresentado na Figura 5.5. No primeiro passo, é caracterizado o contexto do projeto. O gerente de projeto informa o contexto do projeto, atribuindo valores para cada informação de contexto que correspondam ao contexto atual do projeto de desenvolvimento.

Com base no contexto do projeto e na LPS criada, o mecanismo de composição da COMPOOTIM é capaz de identificar as situações e regras de contexto que se aplicam ao projeto. Dessa forma, as expressões que representam cada uma das situações e regras de contexto são avaliadas, com o objetivo de identificar aquelas que foram consideradas como verdadeiras para o projeto em questão.

Há também algumas decisões sobre configuração de variabilidades e opcionalidades que o gerente do projeto deverá tomar com base na sua percepção.

Usando as regras de contexto aplicáveis ao projeto e as regras de composição de características definidas previamente, são recortadas ou filtradas as características de processo específicas para o projeto. Para que uma regra seja aplicada, a expressão que representa o seu antecedente deve ser verdadeira. As regras são aplicadas sucessivamente verificando-se a consistência entre os resultados gerados pelo passo atual e os passos anteriores e aplicando as modificações necessárias no conjunto de características selecionadas. Estes filtros também resolvem as variabilidades, removendo os pontos de variação cujas variantes já foram selecionadas pelo gerente de projeto.

Ao final, estes filtros determinam as características que devem ser adicionadas ou removidas para o recorte do processo de acordo com o contexto do projeto. A seleção dessas características também vai influenciar a seleção ou exclusão dos componentes de processo nas etapas futuras.

Depois que os conjuntos de características adicionadas e removidas são definidos, é possível realizar a verificação da consistência das regras, conforme definido por Fernandes (2008).

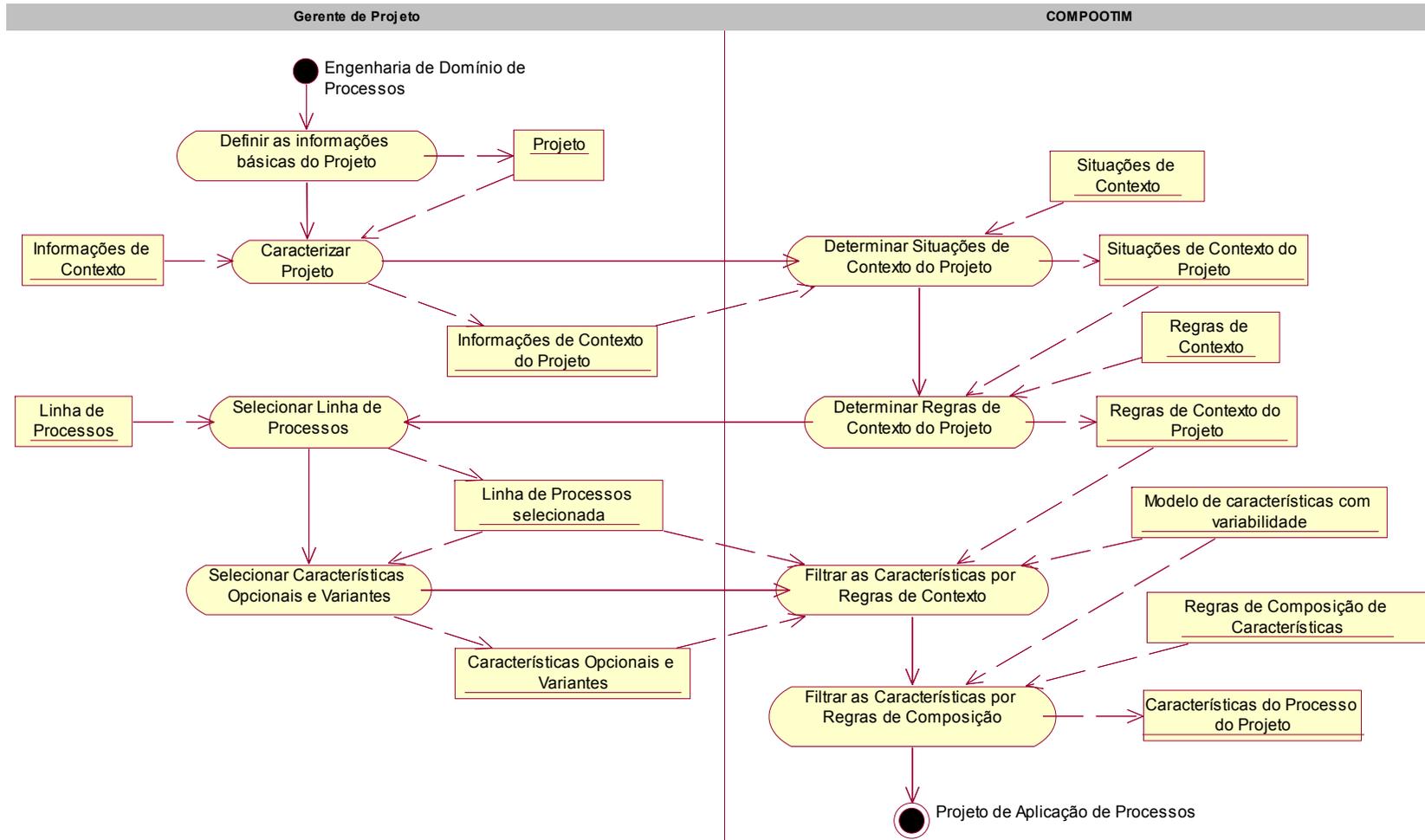


Figura 5.5 – Passos e artefatos da fase de Análise de aplicação de processos

b) Projeto de Aplicação de Processos

Os passos e artefatos resultantes desta fase são detalhados no diagrama apresentado na Figura 5.6. Na fase Projeto de Aplicação de Processos, é feito o recorte dos componentes de processo, a partir do recorte das características já realizado na etapa anterior e aplicando as regras de composição de componentes.

O filtro dos componentes pelas características parte da arquitetura completa de componentes definida na LPS e seleciona apenas aqueles componentes associados às características que foram incluídas para o projeto. Em seguida, são aplicadas as regras de composição dos componentes, que também incluem ou excluem alguns componentes do processo.

O último passo é a otimização da colaboração e o sequenciamento dos componentes. Este passo não é mais realizado pelo mecanismo de composição e sim pelo mecanismo de otimização. Então, ele é detalhado na próxima seção.

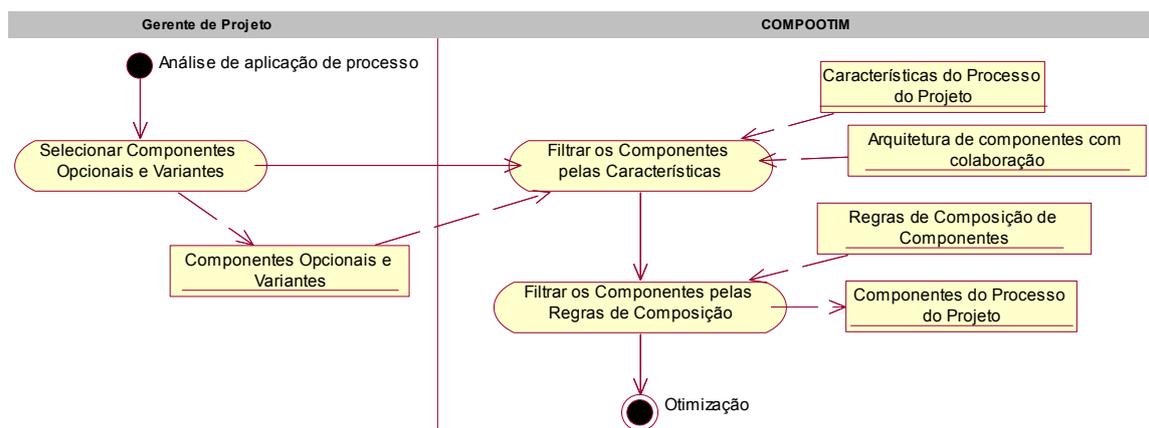


Figura 5.6 – Passos e artefatos da fase de Projeto de aplicação de processos

5.2.1.3. Execução da Otimização

A etapa de otimização da COMPOOTIM segue os passos definidos na Figura 5.7. Nesta etapa, o Gerente de Projeto pode escolher a estratégia de medição da colaboração que deseja que seja aplicada ao seu projeto. Caso ele não saiba fazer essa escolha, a estratégia da mediana será aplicada por *default*. A partir dessa escolha, o mecanismo de otimização irá sequenciar os componentes e otimizar a colaboração, conforme descrito na Seção 5.3.

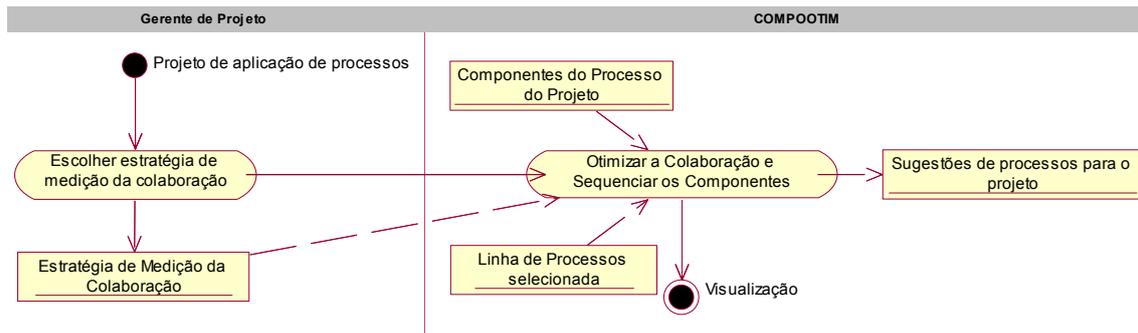


Figura 5.7 – Passos e artefatos da etapa de Otimização

5.2.2. Ferramental de Apoio

Para fornecer o apoio computacional necessário à otimização da colaboração na composição de processos de software, foi desenvolvido na COMPOOTIM um ferramental de apoio que consiste em três mecanismos principais interligados (Figura 5.8).

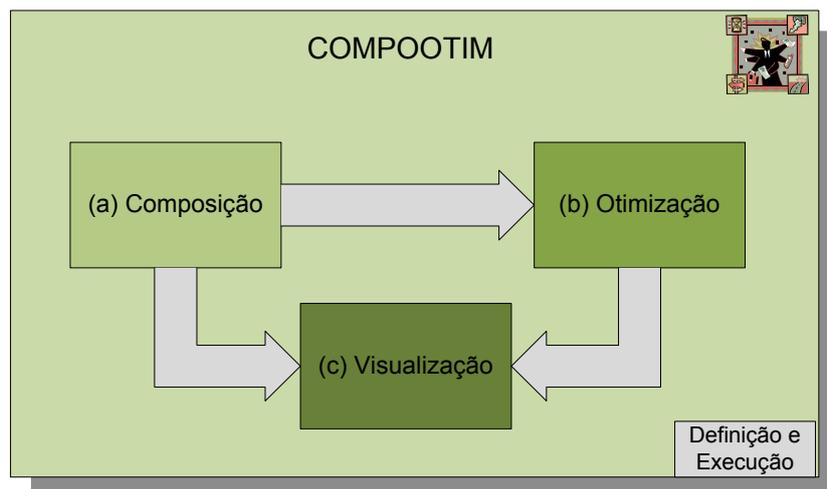


Figura 5.8 – Mecanismos da COMPOOTIM

O primeiro é o **mecanismo de composição** (Figura 5.8a), onde é criada a LPS e iniciado o recorte das características e componentes de processo de acordo com o contexto do projeto. Regras de contexto e composição atuam como filtros, descartando aquelas características e componentes de processos que não se aplicam ao projeto e selecionando aqueles compatíveis com o contexto do projeto.

O **mecanismo de otimização** (Figura 5.8b) é responsável pelo sequenciamento dos componentes bem como pela otimização da colaboração no processo resultante. Ele procura um espaço de solução com vários candidatos de processos resultantes, tentando encontrar o conjunto de parâmetros que maximize a colaboração. O sequenciamento conecta os

componentes anteriormente selecionados, de acordo com as suas interfaces requeridas (artefatos de entrada) e interfaces providas (artefatos de saída). O resultado consiste em um ou mais processos possíveis de serem utilizados no projeto.

Finalmente, o **mecanismo de visualização** (Figura 5.8c) é a camada de apresentação. Ele resume as informações sobre o contexto do projeto e também descreve as sugestões de processos através de uma sequência de componentes e da lista de artefatos gerados.

Todos os mecanismos da COMPOOTIM foram implementados na linguagem Java usando uma arquitetura MVC em uma solução *standalone*. A COMPOOTIM foi documentada através de casos de uso e do diagrama de classes, disponíveis no site¹⁴. Os detalhes sobre as funcionalidades de cada um desses mecanismos são apresentados na Seção 5.5.

5.3. Otimização da Colaboração

A composição de processos tratada neste trabalho é entendida como um problema de otimização devido à necessidade de maximizar a colaboração em uma composição de processos para um determinado contexto. Utilizando como base algoritmos de otimização, busca por soluções que maximizem o nível de colaboração do processo composto para o projeto.

5.3.1. Mecanismo de Otimização

O mecanismo de otimização é responsável por propor ao gerente de projeto um processo que satisfaça as restrições do contexto do projeto e que otimize a colaboração. Ele foi construído utilizando como base métodos de otimização que têm como finalidade buscar soluções que atendam a um conjunto de restrições impostas, para maximizar um determinado fator (PAPADIMITRIOU e STEIGLITZ, 1998). Os métodos de otimização são descritos na próxima seção.

5.3.1.1. Métodos de Otimização

Métodos de otimização podem fornecer soluções para problemas complexos e podem sugerir formas de encontrar soluções aceitáveis em situações onde

¹⁴ Site da COMPOOTIM: <http://reuse.cos.ufrj.br/cdsoft/compootim.html>

soluções perfeitas são teoricamente impossíveis ou praticamente inviáveis (HARMAN e JONES, 2001).

A otimização busca melhorar o desempenho em direção a um ponto ou pontos ótimos (GOLDBERG, 1989). Otimização consiste em encontrar uma ou mais soluções válidas para um determinado problema que correspondem a valores extremos (ou ótimos) de uma ou mais funções que valoram tais soluções.

Cada solução gerada para o problema é conhecida como solução candidata. Se uma determinada solução candidata satisfaz todas as restrições do problema, então ela é chamada de solução viável. As soluções ótimas são soluções viáveis que conduzem a função de *fitness* para o valor ótimo. O conjunto de todas as soluções candidatas é conhecido como espaço de busca. Em alguns casos, encontrar a solução ótima global para um problema pode ser proibitivamente difícil. Nestes casos, pode-se buscar uma solução ótima local que seja a melhor possível de ser encontrada (PAPADIMITRIOU e STEIGLITZ, 1998).

Normalmente, há várias soluções candidatas para um problema de otimização. A otimização consiste em encontrar o conjunto de parâmetros para os quais as funções de *fitness* têm um valor máximo ou mínimo. As funções de *fitness* caracterizam a qualidade da solução. Cada função de *fitness* terá uma solução ótima local própria, potencialmente distinta das soluções ótimas locais geradas para outras funções de *fitness*. As funções de *fitness* ajudam a identificar a técnica de busca mais aplicável e a entender melhor a natureza das soluções candidatas (HARMAN e JONES, 2001).

A qualidade de uma solução é medida de acordo com o valor gerado pela função de *fitness* e a sua proximidade do valor ótimo. Quanto mais próximo do valor ótimo estiver o valor calculado pela função de *fitness* para uma determinada solução candidata, melhor será a sua qualidade.

Existem dois tipos de métodos para resolver um problema de otimização. Os métodos de otimização capazes de encontrar soluções ótimas para os problemas são conhecidos como **métodos exatos**. Alguns exemplos de métodos exatos são: busca sistemática, programação linear e *branch and bound*. As descrições e exemplos de aplicações de cada um destes métodos podem ser encontrados na literatura (HARMAN, 2007, NETTO, 2010).

Entretanto, dependendo da natureza do problema, é impossível encontrar a solução ótima, em tempo exequível, a partir de um método exato que se proponha a avaliar todas as soluções possíveis de um dado problema e selecionar a melhor dentre elas. Estes algoritmos muitas vezes não são aplicáveis a problemas de ES reais, onde existem múltiplos critérios e funções de *fitness* complexas. Nestes casos, uma alternativa é utilizar **métodos heurísticos**.

Segundo Reeves (1993), heurísticas são técnicas que têm por finalidade a busca de soluções boas (próximas da ótima) para problemas complexos cuja solução ótima não pode ser encontrada para todas as instâncias em tempo exequível, pois o espaço de busca cresce em escala exponencial. Consistem na aplicação de regras que tentam guiar a solução para um ótimo local (a melhor solução considerando um subconjunto do conjunto das soluções viáveis para o problema), na esperança de que este ponto também seja o ótimo global (a melhor dentre todas as soluções possíveis).

Estas técnicas também possuem algumas limitações. Em geral, não oferecem garantias de qualidade e viabilidade das soluções geradas, pois nem sempre a solução ótima global será obtida e, muitas vezes, não é possível afirmar quão próxima a solução gerada está da solução ótima. A previsibilidade também é baixa, pois cada execução pode obter resultados diferentes, mas sempre melhores que os anteriores (HARMAN, 2007).

Apesar destes desafios, os métodos heurísticos têm sido aplicados com bastante frequência nas áreas de engenharia, financeira e industrial. Contudo, estas iniciativas ainda são recentes na área de ES.

O termo Engenharia de Software Baseada em Busca (*Search Based Software Engineering - SBSE*) foi cunhado por Harman e Jones (2001) e consiste em uma área de pesquisa que estuda a Engenharia de Software sob a ótica de problemas de busca e otimização, com a aplicação de técnicas heurísticas para buscar soluções aproximadas. Desde então, houve um acentuado crescimento no número de trabalhos nesta linha de pesquisa, em diversas áreas da Engenharia de Software (CLARKE *et al.*, 2003, HARMAN, 2007, HARMAN *et al.*, 2009), tais como análise de requisitos, projeto arquitetural, testes, manutenção, gerência da configuração e gerenciamento de projetos (ALBA e CHICANO, 2007, NETTO, 2010).

A ES se mostrou uma área muito propícia a aplicação de técnicas de otimização baseadas em buscas, devido à natureza dos problemas, onde soluções perfeitas são muitas vezes impossíveis ou impraticáveis, e às propriedades do software. Além disso, SBSE também se mostrou atrativa aos pesquisadores de ES, pois não impõe uma curva de aprendizado agressiva e nem requer anos de aprendizado nas técnicas, fundamentações e nomenclatura dos algoritmos de otimização (HARMAN, 2010).

Para Clarke *et al.* (2003), a análise da aderência da aplicação de uma solução heurística deve considerar os seguintes critérios: (i) o espaço de busca do problema deve ser tão grande (crescimento exponencial) que inviabilize a busca da solução através de um método exato em tempo polinomial; (ii) inexistência de soluções completas, capazes de prover uma solução para o problema para qualquer instância em tempo polinomial, pois não faz sentido tentar resolver um problema que já foi resolvido; (iii) existência de uma função de *fitness* adequada ao contexto do problema; (iv) as soluções candidatas devem ser encontradas em tempos exequíveis, isto é, o tempo de resposta do algoritmo não pode comprometer a sua continuidade. Se um determinado problema atende a estes quatro critérios, então um algoritmo heurístico pode ser adequado para buscar uma solução aproximada.

Muitos problemas relacionados ao desenvolvimento de software podem ser encarados como problemas de otimização, onde há muitas alternativas a selecionar com base em um conjunto de restrições e as soluções próximas do ótimo são suficientes desde que se encaixem em um limite de tolerância aceitável (HARMAN *et al.*, 2009). Assim, métodos de otimização/busca heurística são utilizados para encontrar boas soluções para estes problemas.

Por exemplo, em Karlstrom e Runeson (2002) é proposta uma abordagem de otimização no que diz respeito à introdução de *Extreme Programming* (XP) nas organizações e à seleção de suas práticas, considerando a facilidade de adoção e o potencial efeito benéfico de cada prática. Um problema similar, de alocação de desenvolvedores em tarefas de correção de *bugs* e o sequenciamento destas tarefas, foi tratado por Netto (2010) utilizando algoritmos genéticos.

Seguindo a ideia de que problemas da ES podem ser formulados como problemas de busca e otimização, na próxima seção é apresentada a

modelagem do problema de composição como um problema de otimização.

5.3.2. Modelagem do Problema de Composição do Processo

Segundo Harman (2007), existem dois ingredientes principais para aplicação de otimização aos problemas de ES: escolher a representação do problema e definir as funções de *fitness* para avaliação das soluções. Atendendo a estes pré-requisitos, é possível aplicar técnicas de otimização a outras áreas de ES, além daquelas que têm sido tradicionalmente exploradas, e obter resultados interessantes e potencialmente importantes (HARMAN et al., 2009).

Assim, nesta seção, é apresentada a definição detalhada do problema de otimização que se quer resolver neste trabalho: como compor o processo para o projeto a partir da LPBC, de forma a otimizar a colaboração? (MAGDALENO, 2010c).

Seguindo a sistemática da COMPOOTIM, o Problema de Composição do Processo do Projeto (PCPP) inicia na fase de Projeto de Aplicação do Processo, quando começa o recorte dos componentes de processo a partir da LPS. A visão geral das etapas que compõem o problema de otimização PCPP é apresentada pela Figura 5.9.

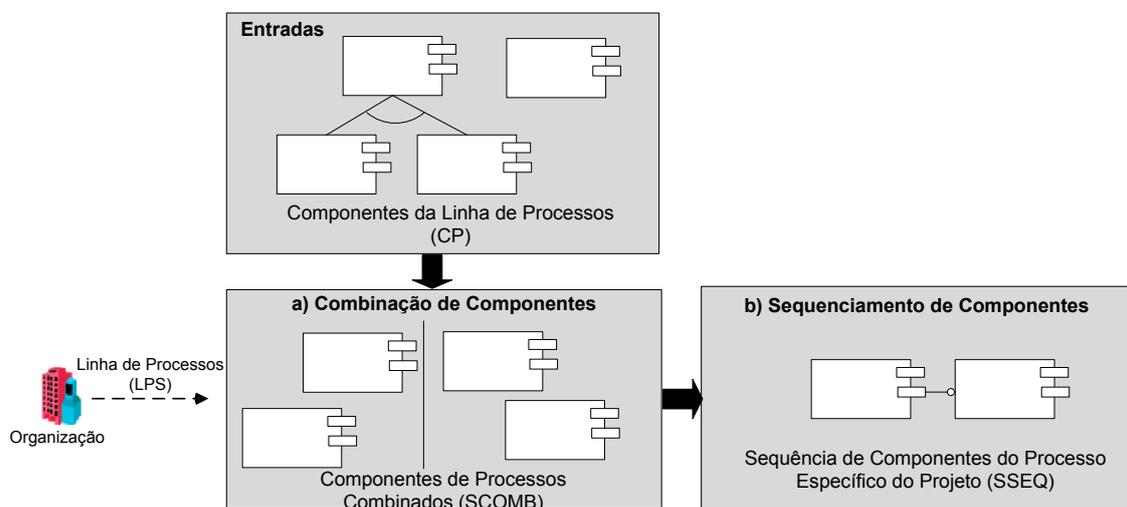


Figura 5.9 – Resumo das etapas do PCPP

A primeira etapa do PCPP envolve a criação das combinações possíveis dos componentes de processos que satisfaçam os resultados esperados pela LPS (Figura 5.9a). Estas combinações serão avaliadas quanto a sua viabilidade de produzir um processo que faça sentido. Em seguida, é feita a otimização da colaboração considerando-se o potencial de colaboração de cada componente, para que sejam escolhidas as melhores combinações. Por

fim, será composto o processo para o projeto através do encaixe dos componentes selecionados respeitando-se os artefatos providos ou requeridos pelas suas interfaces (Figura 5.9b).

As seções seguintes apresentam a modelagem formal de cada uma das fases do problema de otimização PCPP. A modelagem inicial do problema está descrita em (MAGDALENO, 2010c, MAGDALENO *et al.*, 2010a), mas desde então foi evoluída. Esta modelagem do problema foi útil para o entendimento da natureza do problema e para apontar as técnicas de otimização mais indicadas para serem implementadas.

5.3.2.1. Combinação dos componentes de processos

A primeira etapa (Figura 5.9a) corresponde à combinação dos componentes de processo. Ela recebe como entrada um conjunto de componentes de processo compatíveis com o projeto, de acordo com suas informações de contexto (SCP), e gera como resultado uma lista de potenciais combinações destes componentes. Esta etapa pode ser modelada da seguinte forma:

- Seja LPS o conjunto das linhas de processos de software. Cada linha de processos $LPS_i \in LPS$ possui um nome e descrição.

$$LPS = \{ LPS_i \}$$

$$LPS_i = [nome_i, descricao_i]$$

- Seja ED o conjunto de todas as entradas disponíveis na LPS. Cada entrada disponível $ED_i \in ED$ possui um nome.

$$ED = \{ ED_i \}$$

$$ED_i = [nome_i]$$

- Seja PE o conjunto de todos os produtos esperados na LPS. Cada produto esperado $PE_i \in PE$ possui um nome.

$$PE = \{ PE_i \}$$

$$PE_i = [nome_i]$$

- Seja CP o conjunto de componentes de processos filtrados da arquitetura da LPS após o recorte das características. Cada componente de processo $CP_i \in CP$ é composto por um nome, um potencial de colaboração,

opcionalidade, variabilidade, um conjunto de interfaces requeridas (artefatos de entrada) e um conjunto de interfaces produzidas (artefatos de saída).

$$CP = \{ CP_i \}$$

$$CP_i = [\text{nome}_i, \text{pcolab}_i, \text{opc}_i, \text{var}_i, \text{INTRCP}_i, \text{INTPCP}_i]$$

- Seja INTRCP_i um conjunto de interfaces requeridas (artefatos de entrada) de um determinado componente de processo. Cada interface requerida $\text{INTRCP}_{ij} \in \text{INTRCP}_i$ possui um nome.

$$\text{INTRCP}_i = \{ \text{INTRCP}_{ij} \}$$

$$\text{INTRCP}_{ij} = [\text{nome}_{ij}]$$

- Seja INTPCP_i um conjunto de interfaces providas (artefatos de saída) de um determinado componente de processo. Cada interface provida $\text{INTPCP}_{ij} \in \text{INTPCP}_i$ possui um nome.

$$\text{INTPCP}_i = \{ \text{INTPCP}_{ij} \}$$

$$\text{INTPCP}_{ij} = [\text{nome}_{ij}]$$

- Seja COMB o conjunto com todas as combinações de componentes de processos vindas de CP .

$$\text{COMB} = \{ CP_i \}$$

$$CP_i \subseteq CP \wedge \nexists s \subseteq CP \mid s \notin \text{COMB}$$

Para selecionar a lista de todas as combinações de componentes viáveis, é necessário filtrar o conjunto SCOMB e remover as combinações que não satisfaçam os produtos esperados pela LPS ou cujos componentes não tenham as suas interfaces requeridas satisfeitas. Esta seleção é expressa formalmente a seguir:

$$\text{SCOMB} = \{ \text{SCOMB}_i \in \text{COMB} \mid \alpha(\text{SCOMB}_i) \wedge \beta(\text{SCOMB}_i) \}$$

A formulação acima indica que toda combinação selecionada deve satisfazer as condições α e β descritas a seguir:

$$\alpha(t) \leftarrow \forall PE_i \in PE \mid \text{evaluate}(CP_i.\text{INTPCP}, t) = \text{true}$$

onde $evaluate(CP_i.INTPCP, t)$ assume um valor booleano indicando se na combinação que está sendo avaliada todos os produtos esperados pelo processo são satisfeitos pela interface provida de algum componente de processo.

$$\beta(t) \leftarrow \forall CP_i \in CP \mid evaluate(CP_i.INTRCP, t) = true$$

onde $evaluate(CP_i.INTRCP, t)$ assume um valor booleano indicando se na combinação que está sendo avaliada todos os componentes de processo têm as suas interfaces requeridas satisfeitas pelos artefatos gerados pelos demais componentes de processo ou pelas entradas disponíveis da LPS.

5.3.2.2. Sequenciamento dos componentes de processo

A etapa de sequenciamento dos componentes de processo (Figura 5.9b) se baseia nas interfaces (artefatos) providas e requeridas. Este sequenciamento é iniciado com um vetor do tamanho da quantidade de componentes que podem participar do processo, onde a posição i recebe valor 0 se o componente não participa do processo e recebe valor 1 se o componente compõe o processo. O sequenciamento recebe como entrada os conjuntos ED e PE (descritos na seção anterior). Como saída do sequenciamento é gerada a ordenação dos componentes e uma lista dos artefatos produzidos pelos componentes presentes no processo.

Todos os componentes que precisam ser avaliados são colocados em uma lista de pendências (LAP) que é percorrida até que esteja vazia (o que representa que um processo válido foi composto) ou que se verifique que a combinação de componentes não é válida. Além disso, todos os artefatos disponíveis em ED são armazenados em uma lista LAC, que vai mantendo a lista dos artefatos produzidos pelo processo.

Os componentes são removidos da lista LAP em rodadas sucessivas que verificam se as interfaces requeridas daquele componente são satisfeitas com os artefatos presentes em LAC. Cada componente satisfeito é guardado na lista de resolvidos (LAR) e adiciona as suas interfaces providas também em LAC para que outros componentes consecutivos possam ser satisfeitos.

Ao final, caso todas as pendências de LAP tenham sido resolvidas, o processo ordenado está indicado em LAR e a lista LAC indica todos os artefatos que foram produzidos por esta opção do processo.

Este sequenciamento é feito para as diferentes opções de processo. Em seguida, cada uma das opções do processo é submetida às funções de *fitness* (detalhadas na próxima seção) para o cálculo da colaboração em cada uma delas e a busca pelas melhores opções. Ao final, as melhores opções de processo encontradas estão disponíveis e podem ser consultadas pelo gerente de projeto no mecanismo de visualização de processos (Seção 5.5.3).

5.3.2.3. Funções de *fitness*

As funções de *fitness* caracterizam o que é considerada uma boa solução e ajudam a compreender melhor a natureza das soluções candidatas. Como o problema deste trabalho de pesquisa envolve a otimização da colaboração na composição do processo, este aspecto da colaboração pode ser usado como função de *fitness* para determinar fatores de qualidade para a solução do problema. Assim, foi estabelecida a seguinte função de *fitness*:

$$(m1 == 1) ? \beta : 0$$

Esta formulação indica que se $m1$ for igual 1, deve ser atribuído o valor de β ao *fitness* do processo. Caso contrário, deve ser atribuído valor zero.

$m1$ corresponde ao % dos artefatos finais gerados pelo processo:

$$0 \leq n / PE \leq 1$$

Onde n corresponde ao número de artefatos produzidos pela combinação de componentes avaliada e PE representa o número de produtos esperados pela LPS.

Desta forma, combinações de componentes inválidas, ou seja, que não satisfaçam completamente todos os produtos esperados pela LPS são punidas com um *fitness* igual a zero. Para aquelas que satisfazem a LPS, é calculado o valor de β de acordo com a estratégia de medição da colaboração escolhida entre aquelas definidas na Seção 5.1.2 (otimista, pessimista, mediana e acumulativa).

5.3.2.4. Análise do espaço de busca

Para caracterizar o espaço de busca é importante analisar a sua complexidade e tamanho. Isso ajuda a entender a dificuldade do problema e a identificar a técnica mais adequada para ser aplicada na solução do problema.

A função “complexidade do tempo” de um algoritmo expressa o maior tempo necessário (pior caso) para solucionar um determinado problema. Em

geral, esta função é representada na forma $O(n)$, onde n significa o tamanho da instância do problema. Algoritmos de tempo polinomial são aqueles cuja função complexidade do tempo pode ser definida como $O(p(n))$ para uma determinada função polinomial p . Os demais algoritmos, cujos tempos de execução não podem ser delimitados, são conhecidos como algoritmos de tempo exponencial. No caso dos algoritmos exponenciais, a solução exata para instâncias grandes do problema não pode ser encontrada em tempo exequível. Nestes casos, devem ser utilizados algoritmos que executam em tempo polinomial para encontrar soluções aproximadas para o problema (NETTO, 2010).

No problema PCPP, a complexidade é influenciada pelo número de componentes de processos (p) e de rodadas até que as suas interfaces requeridas sejam satisfeitas (a). Considerando estes fatores, a complexidade do problema é exponencial: $O(a * 2^p)$. No entanto, este é um limite teórico e não real, pois não leva em consideração os componentes conflitantes entre si e que não poderiam ser utilizados em um mesmo processo. Quando as restrições são adicionadas na formulação do problema, o esforço necessário para encontrar as soluções tende a decrescer, devido ao número de caminhos que não atendem às restrições. Desta forma, o número de possíveis componentes de processos diminui para p' , onde $p' < p$. Assim, na prática, o espaço de busca cresce proporcionalmente a $O(a * 2^{p'})$.

Este crescimento do espaço de busca em relação ao tamanho da instância inviabiliza a aplicação de métodos exatos, pois a solução não pode ser encontrada em um tempo exequível. Assim, uma solução utilizando métodos heurísticos foi a opção mais indicada para implementar a solução do PCPP.

Em um cenário onde exista uma LPS complexa, com dezenas de componentes e suas respectivas interfaces de conexão com outros componentes, o número de combinações possíveis entre eles pode ser muito grande. Note que a inclusão de alguns poucos componentes em uma LPS pode fazer o número de possíveis processos aumentar muito. Assim, o problema de otimização pode não ser resolvível em tempo computacionalmente aceitável, se todas as possíveis combinações entre os componentes tiverem que ser examinadas. Para equacionar isso, são usados os algoritmos de otimização descritos na próxima seção.

5.3.3. Algoritmos de Otimização

Os algoritmos de otimização baseados em buscas (algoritmos heurísticos) utilizam diferentes estratégias para identificar soluções ótimas ou próximas do ótimo para um determinado problema. Estas estratégias avaliam uma série de soluções candidatas, sem analisar todas as alternativas possíveis. Tal característica é particularmente útil quando a análise de todas as alternativas não puder ser realizada em tempo viável.

Os principais algoritmos de otimização utilizados na SBSE são: o *Random Search* (RS), *Hill Climbing* (HC), *Simulated Annealing* (SA) e Algoritmos Genéticos (GA – do inglês *Genetic Algorithm*). Outros algoritmos também são utilizados em menor escala como o Colônia de Formigas e Nuvem de Partículas (HARMAN *et al.*, 2009).

Neste trabalho, adotou-se a estratégia sugerida por Clarke *et al.* (2003) que sugerem que se comece pela técnica mais simples para verificar se os resultados são encorajadores antes de investir em outras técnicas mais sofisticadas que podem representar um esforço adicional desnecessário. Assim, o RS e o HC foram os algoritmos utilizados neste trabalho e apenas eles serão aprofundados nas próximas seções. Para mais detalhes sobre os demais algoritmos e sua aplicação na ES, sugere-se consultar o *survey* de Harman *et al.* (2009) sobre o assunto.

a) *Random Search* (RS)

O *Random Search* (RS) é um algoritmo de busca não sistemática. O RS é um algoritmo iterativo que busca soluções para um problema por amostragem aleatória, armazenando a melhor solução encontrada até o momento.

O algoritmo começa com uma solução candidata, gerada de forma aleatória, que é avaliada pela função de *fitness*. A cada iteração, uma nova solução aleatória é gerada e avaliada pela mesma função de *fitness*. Se a nova solução possui *fitness* melhor do que a solução atual, ela é eleita como solução atual. O processo se repete até que um determinado número de soluções aleatórias tenha sido gerado. Este algoritmo executa até que o número de avaliações concedido acabe.

b) Hill Climbing (HC)

O *Hill Climbing* (HC) é o algoritmo de busca local mais conhecido e aplicado em problemas de SBSE. O algoritmo começa com uma solução candidata, gerada de forma aleatória, que é avaliada pela função de *fitness*. A cada iteração, um conjunto de "soluções vizinhas" à solução atual são consideradas e avaliadas. A definição de "solução vizinha" depende de cada aplicação em que o algoritmo é utilizado, mas em geral elas são geradas por um mecanismo simples (uma variante da solução original). Se alguma solução vizinha possui *fitness* melhor do que a solução atual, ela é eleita como solução atual e o processo recomeça. Se nenhuma solução vizinha possui *fitness* melhor do que a solução atual, a busca termina e uma solução possivelmente ótima é encontrada.

5.4. Acompanhamento da Colaboração

O acompanhamento da colaboração diz respeito a monitorar qual é o nível real de colaboração no processo, verificar se a colaboração está de fato acontecendo conforme o planejado e avaliar os resultados alcançados com a colaboração no processo. Neste momento, o gerente de projeto precisa ser munido de informações sobre a execução do projeto para tomar decisões e possíveis ações corretivas.

5.4.1. Características de colaboração em redes sociais de desenvolvimento de software

Das propriedades de análise de redes sociais (ARS) apresentadas no Capítulo 2, algumas possuem maior potencial para explicitar a colaboração e foram detalhadas em (MAGDALENO *et al.*, 2010b, SANTOS *et al.*, 2011). A partir da interpretação e composição destas propriedades, Santos *et al.* (2010, 2009) sugerem um conjunto de características ou padrões de colaboração em equipes de desenvolvimento de software. Nesta proposta, a ideia principal é que as informações obtidas através das propriedades de ARS podem ser associadas aos níveis de maturidade em colaboração do CollabMM (MAGDALENO *et al.*, 2009a), para estabelecer a característica de colaboração de uma determinada rede social. A Tabela 5.8 resume as características de

colaboração em redes sociais de desenvolvimento de software, focadas no aspecto da coordenação entre seus membros.

A característica de *coordenação absoluta* está relacionada ao nível planejado do CollabMM. As redes que possuem esta característica apresentam um nó/ator forte o suficiente (líder), com potencial para dominar toda a rede. Como se trata de uma rede centralizada, os líderes atuam como pontes entre os demais nós da rede. Entretanto, a densidade da rede é baixa, pois a quantidade de interações entre os atores é pequena.

Redes caracterizadas com coordenação múltipla apresentam os primeiros sinais de descentralização e, portanto, existem alguns nós centrais e intermediários estabelecendo ligações entre pequenos subgrupos dentro da mesma rede.

Nas redes de coordenação distribuída, ocorre a ausência de nós centrais e intermediários na rede, pois os relacionamentos entre os nós tendem a ser distribuídos igualmente na rede. A densidade da rede é considerada alta, podendo atingir a chamada densidade máxima da rede que representa o grau máximo de colaboração (SANTOS *et al.*, 2010).

Tabela 5.8 – Características de coordenação em redes sociais de desenvolvimento de software (SANTOS *et al.*, 2010)

	Coordenação Absoluta Nível 2 - Planejado	Coordenação Múltipla Nível 3 - Perceptivo	Coordenação Distribuída Nível 4 - Reflexivo
Densidade da rede	Baixa (0,01% a 30,00%)	Baixo-média a médio-alta (30,01% a 70,00%)	Alta (70,01% a 100,00%)
Centralidade de grau da rede	Alta (70,01% a 100,00)	Médio-alta a baixo-média (30,01% a 70,00%)	Baixa (0,00% a 30,00%)
Centralidade de intermediação da rede	Alta a médio-alta (50,01% a 100,00%)	Baixo a baixo-média (0,00% a 50,00%)	Baixa (0,00% a 30,00%)

Estudos exploratórios iniciais em projetos de desenvolvimento de software livre foram realizados com o objetivo principal de verificar a existência destas características de colaboração. Os resultados destes estudos, apresentados em Santos *et al.* (2010, 2009), apontam para a viabilidade

destas características como instrumento para oferecer percepção da colaboração.

5.4.1.1. Cenário de Uso das Informações de Colaboração

O trabalho de Santos *et al.* (2010, 2009) fornece um importante insumo para a presente tese, que se vale das características de colaboração identificadas para planejar o nível de colaboração necessário e acompanhar o nível atual de colaboração nos projetos de desenvolvimento de software.

Mesmo através de um exemplo simples, como o que será apresentado a seguir, é possível perceber que a combinação dos instrumentos CollabMM e redes sociais pode fornecer informações úteis para o gerente de projeto tomar decisões sobre o futuro do projeto e melhorar o nível de colaboração (MAGDALENO *et al.*, 2012b).

Para ilustrar o planejamento da colaboração, pode-se considerar como exemplo um projeto de desenvolvimento de software que envolve uma equipe distribuída geograficamente e visa criar um produto inovador. Devido à novidade do produto, também tem requisitos voláteis, que mudam com frequência. Por causa desta instabilidade de requisitos, a documentação se torna obsoleta rapidamente e a necessidade de compartilhar o conhecimento tácito entre os membros da equipe é intensificada. Neste cenário, o projeto exigiria um alto nível de colaboração.

Este projeto se beneficiaria se a colaboração atingisse o nível Reflexivo do CollabMM (Seção 5.1.1.1). A característica de coordenação distribuída (Tabela 5.8) representa esse nível. Então, uma rede social similar a apresentada na Figura 5.10a, que tem alta densidade e onde os relacionamentos entre os nós tendem a ser igualmente distribuídos, é esperada.

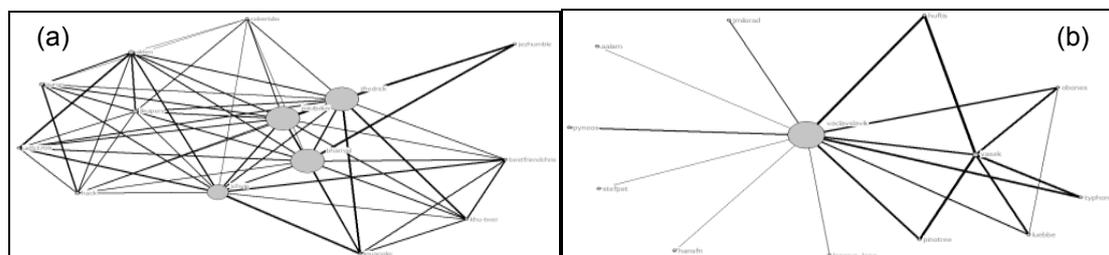


Figura 5.10 – Exemplos de redes sociais – Planejada (a) e Atual (b)

Contudo, durante a execução do projeto, a rede social atual começa a parecer com a apresentada na Figura 5.10b. Nesta rede centralizada, existe

uma forte liderança de um único nó, que controla as tarefas e o fluxo de informações. Baseado nesta informação, é possível rever, usando os níveis e práticas de colaboração do CollabMM, como atingir os resultados de colaboração planejados para o projeto. Neste caso, é possível incluir, alterar ou excluir um ou mais componentes de processo de forma que o processo composto para o projeto ganhe novas características necessárias para estimular a colaboração.

Na prática, para que este acompanhamento da evolução da rede social seja possível, é necessário contar com o uso de um ferramental de apoio que permita visualizar as mudanças que vão ocorrendo. Assim, o gerente de projeto é capaz de perceber, refletir e interferir no trabalho que está sendo realizado. A ferramenta EvoTrack-SocialNetwork cujo objetivo é acompanhar a colaboração em projetos de desenvolvimento de software é o tema da próxima seção.

5.4.2. Ferramenta EvoTrack-SocialNetwork

A EvoTrack-SocialNetwork¹⁵ é uma extensão da ferramenta EvoTrack (CEPEDA *et al.*, 2008, WERNER *et al.*, 2011b), que constitui um dos projetos do Grupo de Reutilização de Software¹⁶. A EvoTrack é uma abordagem para capturar e visualizar o ciclo de evoluções de um projeto de software. Ela extrai, periodicamente, as informações do projeto de uma fonte de dados e, depois de realizar processamentos e transformações, apresenta o design correspondente ao projeto. A ferramenta foi criada como um *plug-in* do ambiente de desenvolvimento Eclipse¹⁷ e sua arquitetura flexível permite a combinação de diferentes conectores de fontes de dados e conectores de visualização.

A EvoTrack foi escolhida como ponto de partida para a construção da ferramenta de redes sociais, pois já oferecia uma infraestrutura inicial para a extração e atualização de dados, recursos de visualização e a funcionalidade para análise de métricas, além de ter sido desenvolvida pelo mesmo grupo de pesquisa no qual este trabalho de pesquisa está inserido.

5.4.2.1. Requisitos para Ferramentas de Redes Sociais

A partir da análise das contribuições e limitações das ferramentas (apresentadas no Capítulo 2), da observação das propostas existentes para

¹⁵ Site EvoTrack-SocialNetwork: <http://reuse.cos.ufrj.br/evoltrack/socialnetwork/>

¹⁶ Site Grupo de Reutilização: <http://reuse.cos.ufrj.br>

¹⁷ Site Eclipse: <http://www.eclipse.org/>

ARS no desenvolvimento de software (também apresentadas no Capítulo 2) e dos objetivos do presente trabalho de pesquisa, foi possível estabelecer uma lista dos principais requisitos desejáveis para uma ferramenta de redes sociais. Os requisitos foram separados nas categorias mineração (REQM), visualização (REQV) e análise (REQA) (MAGDALENO *et al.*, 2010b, 2010c) (Tabela 5.9).

A Tabela 5.9 classifica as ferramentas de redes sociais analisadas no Capítulo 2 de acordo com os requisitos apresentados. Os campos preenchidos com  indicam que o requisito em questão é totalmente coberto pela ferramenta. Os campos preenchidos com  indicam que o requisito não é atendido pela ferramenta. Os campos preenchidos com  indicam que não foi possível avaliar se o requisito é atendido pela ferramenta através da documentação disponível.

O fato de um requisito não ser atendido por uma determinada ferramenta não significa que essa ferramenta é pior que as demais, pois para o seu objetivo esse requisito pode ser irrelevante. Esses requisitos, apesar de não formarem um conjunto completo nem necessariamente suficiente, servem como um guia para a elaboração de uma ferramenta que se proponha a ser abrangente e que traga novas contribuições em relação às ferramentas atualmente disponíveis na literatura.

Após a análise da Tabela 5.9, é possível concluir que nenhuma das ferramentas de redes sociais estudadas atende a todos os requisitos. Isto motivou a criação da EvolTrack-SocialNetwork para atender de forma mais adequada a necessidade de acompanhamento da colaboração através da implementação destes requisitos.

Tabela 5.9 – Requisitos para ferramentas de redes sociais

Requisito	Descrição	Ariadne	Augur	MiSoN	OSSNet work	Pajek	RaisAwa re	Sargas	SVNNAT	UCINET	Visone
REQM1	Atualizar os dados das redes sociais de forma constante.	?	?	?	×	×	×	×	×	×	×
REQM2	Minerar dados de diferentes fontes de informações de projetos de desenvolvimento de software.	×	×	×	✓	×	×	✓	×	×	×
REQV3	Oferecer a visualização da rede técnica, através de um grafo, onde os nós são as classes e as arestas representam as dependências existentes entre elas.	✓	×	×	×	×	✓	×	×	×	×
REQV4	Oferecer a visualização da rede sócio-técnica, através de um grafo, onde existem nós de classes e desenvolvedores e arestas entre eles indicam criação ou edição.	✓	✓	×	×	×	✓	×	×	×	×
REQV5	Oferecer a visualização da rede técnica e sócio-técnica com diferentes granularidades: por classe ou por pacote.	✓	×	×	×	×	×	×	×	×	×
REQV6	Oferecer a visualização da rede social, através de um grafo, com as relações de dependência e conflito entre os atores.	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
REQV7	Destacar, na visualização das redes, o tamanho do nó de acordo com o seu número de contribuições.	×	✓	×	×	×	×	×	×	×	×
REQV8	Destacar, na visualização das redes, a informação de temporalidade, mudando o nível de transparência de acordo com o quão recente é uma determinada contribuição.	×	✓	×	×	×	×	×	×	×	×
REQV9	Oferecer a visualização da evolução da rede ao longo do tempo.	✓	×	×	✓	×	×	×	×	×	×
REQA10	Calcular as propriedades de centralidade de grau, centralidade de intermediação e centralidade proximidade de cada nó e da rede como um todo (de acordo com as propriedades de ARS apresentadas no Capítulo 2).	×	×	×	✓	✓	✓	✓	×	✓	✓
REQA11	Calcular a propriedade de densidade da rede (de acordo com as propriedades de ARS apresentadas no Capítulo 2).	×	×	×	✓	✓	✓	×	×	✓	✓

5.4.2.2. Arquitetura da EvolTrack-SocialNetwork

A ferramenta EvolTrack-SocialNetwork, proposta como parte da solução deste trabalho, tem como objetivo atender aos requisitos, identificados na seção anterior, para ferramentas de redes sociais.

A arquitetura da EvolTrack-SocialNetwork (Figura 5.11) (MAGDALENO *et al.*, 2010c, VAHIA *et al.*, 2011) é composta por três módulos: mineração, visualização e análise de redes sociais. Além disso, como foi aproveitada a infraestrutura da EvolTrack, também são utilizados os seus componentes. O **EvolTrack-Kernel** é responsável por orquestrar o fluxo de informações entre conectores, transformadores e visualizadores, acompanhar as mudanças de versões e manter a persistência dos modelos criados.

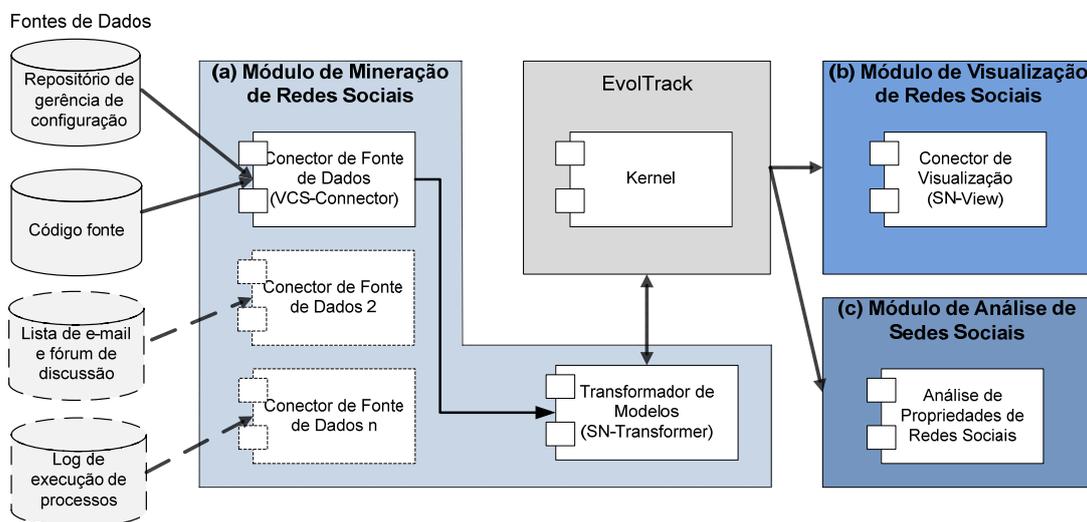


Figura 5.11 – Visão geral da arquitetura da EvolTrack-SocialNetwork

O **módulo de mineração de redes sociais** (Figura 5.11a) recebe as informações extraídas pelos conectores de fontes de dados (que obtêm informações de uma fonte de dados do projeto) e pretende atender aos requisitos REQM1 e REQM2. O REQM1 é automaticamente satisfeito, pois a EvolTrack atualiza os dados sempre que existem modificações nas fontes de dados. Por outro lado, o REQM2 não é completamente satisfeito, visto que atualmente a EvolTrack possui conectores de fontes de dados apenas para ferramentas de gerência de configuração e ambientes de desenvolvimento. Entretanto, a EvolTrack já oferece a infraestrutura necessária para facilitar a criação de novos conectores. Neste caso, os conectores de fonte de dados que precisam ser construídos ou acoplados a EvolTrack-SocialNetwork são: lista de e-mail e fórum de discussão.

Neste módulo foi criado um transformador, o *SNTransformer* (Figura 5.11a). Este recebe as informações extraídas pelos conectores de fontes de dados e é responsável pela marcação do modelo com informações relativas às redes sociais, tais como: autoria, data de modificação de artefatos, etc.

O **módulo de visualização de redes sociais** (Figura 5.11b) recebe as informações coletadas, transformadas e processadas pelo *kernel*. Este módulo tem por objetivo analisar as informações recebidas, gerar e apresentar as redes correspondentes a cada versão do projeto. Então, se concentra em atender aos requisitos REQV3 a REQV9.

Por fim, o **módulo de análise de redes sociais** (Figura 5.11c) tem o foco nos requisitos REQA10 a REQA11. Este módulo é responsável por calcular as propriedades de ARS de cada rede. Além de coletar suas informações, ele também calcula os valores de cada métrica e sinaliza alguma modificação visual que possivelmente deve ser feita nos grafos.

Em relação à implementação, a ferramenta foi desenvolvida em Java usando o framework JUNG¹⁸ para a geração dos grafos. O JUNG é uma biblioteca de software livre que fornece uma linguagem comum e extensível para a visualização e análise de dados que podem ser representados como um grafo ou rede, através de vértices e arestas. O JUNG conta com várias ferramentas para manipulação e geração de grafos, com vasta e documentada API (*Application Programming Interface*) e algoritmos de análise de grafos. Esta biblioteca também possui suporte para transformadores de visualização, responsáveis por efeitos visuais nos nós ou arestas.

5.4.2.3. Funcionalidades da EvoTrack-SocialNetwork

A EvoTrack-SocialNetwork (VAHIA *et al.*, 2011) tem como objetivo prover informações de colaboração que ajudem a perceber o nível de colaboração do projeto. Para o gerente de projeto, a ferramenta fornece informações úteis para acompanhar e monitorar o andamento do projeto, os conflitos e necessidades de comunicação. Ela abre a possibilidade de gerenciar a colaboração durante a execução do projeto através da tomada de decisão e da interferência no trabalho que está sendo realizado através de mudanças nos papéis e responsabilidades dos atores do projeto.

¹⁸ Site JUNG (*Java Universal Network/Graph Framework*): <http://jung.sourceforge.net/>

Para os desenvolvedores, a ferramenta é capaz de oferecer percepção sobre os trabalhos dos outros, fornecendo contexto às suas próprias atividades. Isso facilita o entendimento das dependências existentes entre os trabalhos dos desenvolvedores e permite que sejam capazes de lidar com o impacto de ações interdependentes (SOUZA e REDMILES, 2011).

A visão geral da EvoTrack-SocialNetwork é apresentada na Figura 5.12, que utiliza como exemplo uma rede extraída do projeto de software livre Floggy¹⁹. Cada aba na *view* principal (Figura 5.12a) corresponde a uma rede (técnica, sócio-técnica e social). No painel de opções de visualização (Figura 5.12b), é possível aplicar filtros, transparência e etc. Acima deste painel, é possível escolher qual modelo se quer visualizar na lista de projetos (Figura 5.12c). Na parte inferior esquerda da tela, há a *SatelliteView* (Figura 5.12d), onde é apresentada uma visão total do grafo em miniatura com destaque para a área visualizada. Finalmente, abaixo do grafo (Figura 5.12e), há um painel de controle para a navegação pelas diferentes versões das redes e um painel de análise das métricas das redes (Figura 5.12f).

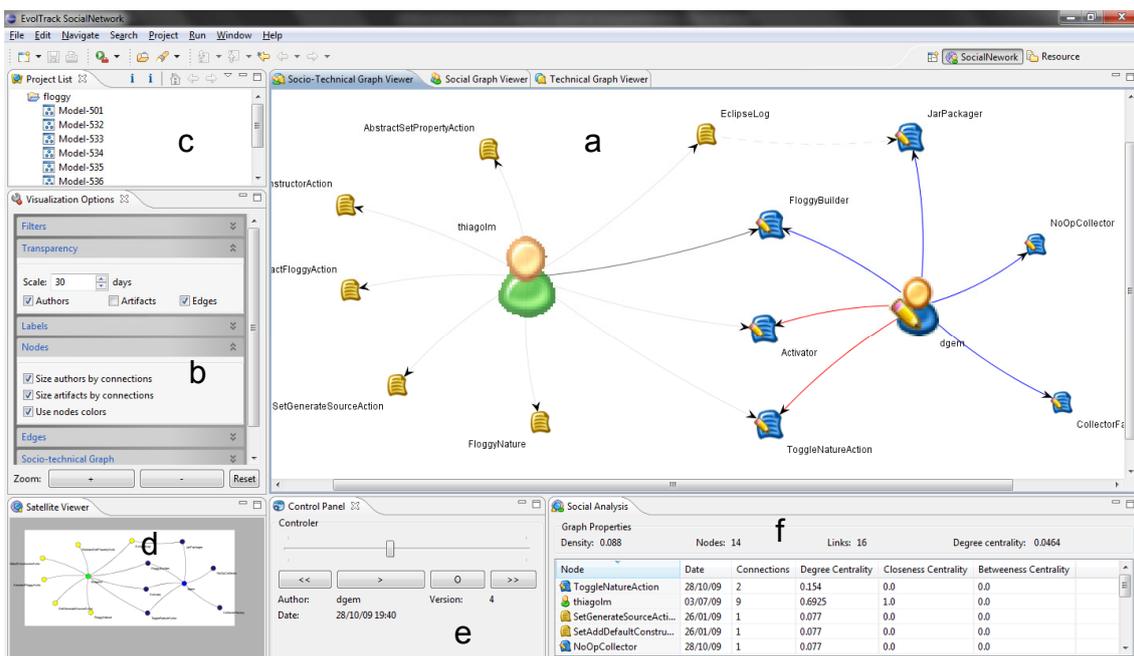


Figura 5.12 – Visão geral da interface da EvoTrack-SocialNetwork usando o projeto Floggy

a) Visualização de Redes Sociais

A EvoTrack-SocialNetwork oferece a visualização dos grafos da rede técnica, sócio-técnica e social (TRAINER *et al.*, 2005). Nestas redes, os nós de autores

¹⁹ Site Floggy: <http://floggy.sourceforge.net/>

são representados através de um ícone de boneco e para os nós de artefatos é adotado um ícone de documento.

A **rede técnica** é extraída através da mineração do código fonte a partir de repositórios de gerência de configuração. Nesta rede (Figura 5.13), todos os nós são artefatos e as arestas pontilhadas representam uma relação de dependência, herança ou implementação entre eles. Este grafo permite representar explicitamente as relações de dependência na arquitetura do software e oferece um entendimento de como esta arquitetura evolui ao longo do tempo.

A partir das dependências e atualizações no código-fonte, é possível compreender também as relações entre os desenvolvedores de software responsáveis por determinado artefato. Quando a rede técnica é acrescida desta informação social, ela origina a **rede sócio-técnica**. Esta rede integra a estrutura técnica do projeto (artefatos) com a informação social (desenvolvedores). As arestas contínuas e direcionadas representam uma relação de autoria/edição entre autores e artefatos. As arestas pontilhadas significam que há uma relação de dependência entre os artefatos. Nesta rede, existem dois layouts possíveis:

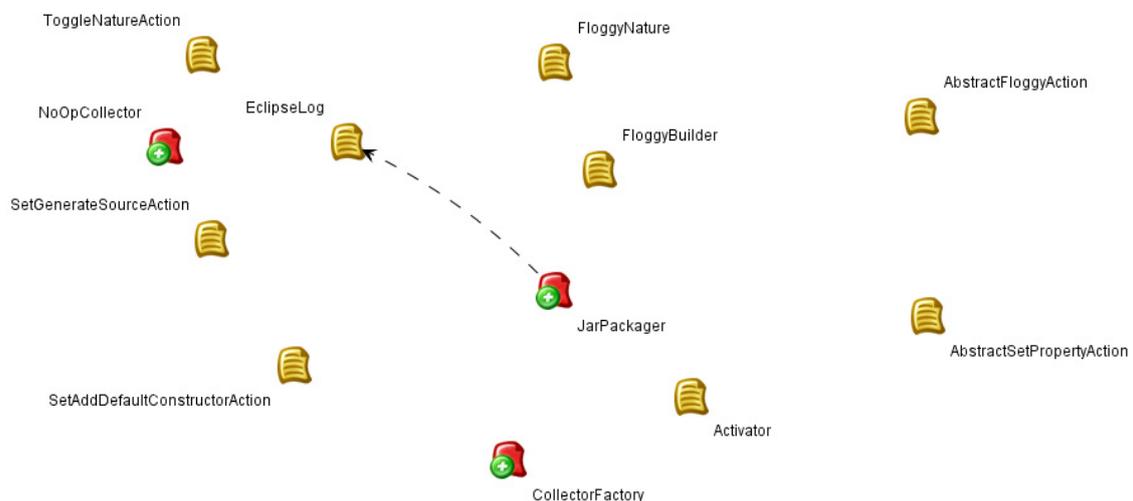


Figura 5.13 – Exemplo de rede técnica do projeto Floggy

- **Global:** os autores são representados por nós únicos, ligados a todos os seus respectivos artefatos (Figura 5.12a);
- **Particionado:** cada nó de artefato é único e tem conectado a ele todos os nós de seus respectivos autores (Figura 5.14).

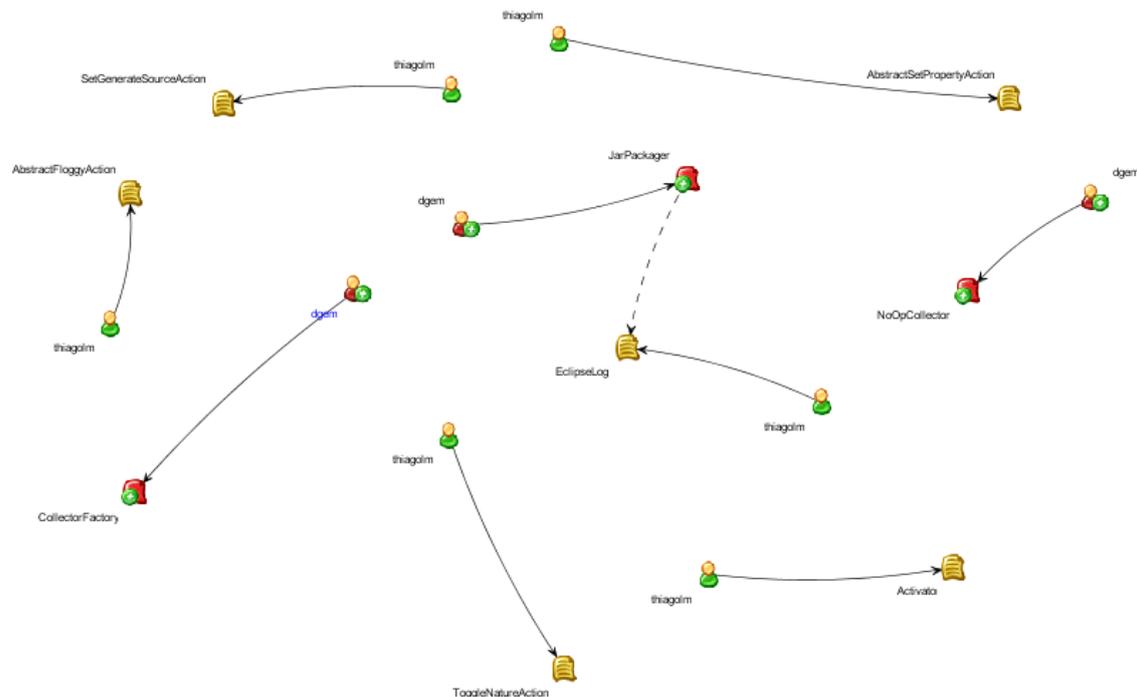


Figura 5.14 – Exemplo de rede sócio-técnica particionada do projeto Floggy

A rede sócio-técnica é útil para identificar os membros da equipe que se envolveram em modificar um determinado artefato. A informação capturada nessa rede pode ser usada, por exemplo, pelo gerente de projeto, para propagar as informações sobre as mudanças necessárias em um determinado artefato. Esta rede também pode ser utilizada para identificar os especialistas em um determinado artefato. O gerente de projeto pode ainda monitorar essa rede para observar o volume de trabalho e detectar artefatos que precisam de outros profissionais trabalhando para dividir o trabalho ou até mesmo refatorar o artefato (DAMIAN *et al.*, 2010).

A partir das relações entre autores e artefatos na rede sócio-técnica, é possível gerar as redes sociais. Na **rede social**, os nós representam unicamente os autores e as relações entre eles. Estas relações entre os desenvolvedores existem devido às dependências no código fonte em que eles estão trabalhando. A rede social de dependência contém os membros que devem se coordenar, pois existem dependências técnicas entre os artefatos em que eles trabalham (DAMIAN *et al.*, 2010). Nesta rede, há também a possibilidade de se observar duas semânticas diferentes:

- **Rede de Dependência:** representa as dependências sociais entre os autores, devido à dependência estrutural entre os respectivos artefatos;
- **Rede de Conflito:** representa os possíveis conflitos entre autores, que

trabalharam ou estão trabalhando no mesmo artefato considerando o contexto de controle de versões. A existência de conflitos no grupo pode implicar em uma baixa participação de um ou mais membros (DAVID, 2004). Quanto mais cedo o gerente de projeto identificá-los, maior a probabilidade de corrigi-los para que o desenvolvimento possa ser bem sucedido.

Com o objetivo de facilitar ainda mais a análise visual de cada rede, foram desenvolvidos ainda alguns recursos avançados de visualização. Em todas as redes, diferentes ícones são usados para mostrar mudanças ou adições nas redes. Os ícones originais mudam de cor e podem ser decorados para ilustrar quando um nó foi adicionado na rede (ícone com sinal +) ou quando ele fez/sofreu alguma modificação (ícone com lápis de edição) (Figura 5.12a). Estes recursos permitem observar o que foi adicionado ou modificado de uma versão para a outra da rede e comparar a evolução dessa rede ao longo das suas versões.

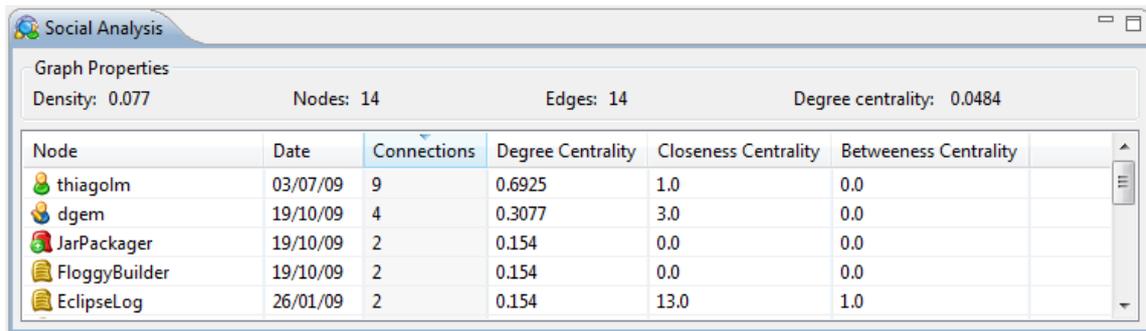
Para observar o comportamento dinâmico destas redes, existe um painel de controle (Figura 5.12e), onde é possível: voltar para o grafo da versão anterior da rede; percorrer os grafos das versões da rede automaticamente e com velocidade constante; ou avançar para o grafo da próxima versão da rede.

A EvolTrack-SocialNetwork oferece **filtros** por: tempo de participação na rede; nome de um determinado nó que se deseje observar com mais interesse; e número de conexões. Uma **escala** de tamanho de acordo com o número de conexões também pode ser aplicada. Neste caso, os ícones de cada nó são aumentados, proporcionalmente, de acordo com o número de relações, conforme pode ser visto na Figura 5.12a. Também é possível optar por aplicar **transparência** aos nós e/ou arestas de acordo com a antiguidade de participação na rede.

b) Análise de Redes Sociais

As propriedades de ARS são calculadas para cada versão do projeto e exibidas em um painel de análise (Figura 5.15). Dados da rede como densidade, número de nós, de arestas e centralidade de grau da rede, são mostrados na parte superior do painel. Abaixo se encontra uma tabela, na qual todos os nós são listados e suas propriedades exibidas, o que permite a ordenação dos nós

de acordo com estas propriedades. A seleção de um nó já o destaca no grafo correspondente.



The screenshot shows a window titled 'Social Analysis' with a 'Graph Properties' section. Below this, a table lists nodes with their respective dates, connection counts, and various centrality metrics.

Graph Properties						
Density: 0.077		Nodes: 14	Edges: 14	Degree centrality: 0.0484		
Node	Date	Connections	Degree Centrality	Closeness Centrality	Betweenness Centrality	
thiagolm	03/07/09	9	0.6925	1.0	0.0	
dgem	19/10/09	4	0.3077	3.0	0.0	
JarPackager	19/10/09	2	0.154	0.0	0.0	
FloggyBuilder	19/10/09	2	0.154	0.0	0.0	
EclipseLog	26/01/09	2	0.154	13.0	1.0	

Figura 5.15 – Painel de análise de redes sociais

Com essa funcionalidade de análise de redes sociais, é possível identificar os elementos que concentram muitos relacionamentos e exercem papéis centrais (*hubs*), nós isolados ou periféricos, nós que são a única ligação entre grupos distintos, aglomerações em pontos isolados ou subgrupos e etc.

Por exemplo, em uma rede sócio-técnica, a propriedade de centralidade de grau pode indicar o nível de contribuição de cada participante. Ao participar de uma atividade cooperativa, cada membro do grupo pode gerar contribuições ou não. Um baixo nível de contribuições pode indicar para o gerente, por exemplo, que os participantes não estão preparados ou estão desmotivados para a atividade. Quanto mais cedo o gerente puder identificar tais índices, maior a probabilidade do grupo atingir o seu objetivo, pois só assim atitudes poderão ser tomadas no sentido de corrigir eventuais falhas (DAVID, 2004).

5.4.2.4. Análise Comparativa

Na Seção 5.4.2.1, foram comparadas e analisadas algumas ferramentas de visualização e análise de redes sociais. Nesta seção, é feita uma análise comparativa entre estas ferramentas e a EvoTrack-SocialNetwork.

Resgatando os principais requisitos esperados para uma ferramenta de rede social, a Tabela 5.10 mostra como a EvoTrack-SocialNetwork satisfaz a estes requisitos. A principal contribuição da ferramenta é permitir o acompanhamento da colaboração. A EvoTrack-SocialNetwork oferece facilidades para navegar de forma dinâmica entre as diversas versões das redes que podem existir em um projeto ao longo do tempo (REQV9). Essa análise histórica é o que permite observar se a rede evoluiu, de acordo com as ações tomadas para estimular a colaboração entre os membros do grupo

(MONCLAR *et al.*, 2011).

Tabela 5.10 – Tabela ferramentas x requisitos com a inclusão da EvoTrack-SocialNetwork

Requisito	Ariadne	Augur	MiSoN	OssNetwork	Pajek	RaisAware	Sargas	SVNNAT	UCINET	Visone	EvoTrack-SocialNetwo
REQM1	?	?	?	×	×	×	×	×	×	×	✓
REQM2	×	×	×	✓	×	×	✓	×	×	×	+/-
REQV3	✓	×	×	×	×	✓	×	×	×	×	✓
REQV4	✓	✓	×	×	×	✓	×	×	×	×	✓
REQV5	✓	×	×	×	×	×	×	×	×	×	+/-
REQV6	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
REQV7	×	✓	×	×	×	×	×	×	×	×	✓
REQV8	×	✓	×	×	×	×	×	×	×	×	✓
REQV9	✓	×	×	✓	×	×	×	×	×	×	✓
REQA10	×	×	×	✓	✓	✓	✓	×	✓	✓	✓
REQA11	×	×	×	✓	✓	✓	×	×	✓	✓	✓

Em relação ao REQM2, considerando as diferentes fontes de informações tratadas pelas ferramentas estudadas, pode-se considerar que ele não é totalmente satisfeito, pois a EvoTrack-SocialNetwork ainda não é capaz de minerar informações de repositórios como e-mail e listas de discussões. Para o presente trabalho, foram selecionadas apenas algumas fontes de informações. Estas fontes de informações elencadas compõem um conjunto inicial, mas outras fontes de informações podem ser identificadas e facilmente agregadas posteriormente, dada a arquitetura flexível da ferramenta.

Quanto ao REQV5, originalmente a EvoTrack não disponibilizava as informações em diferentes granularidades (pacotes, classes, linhas de código e etc.) mas esta possibilidade foi construída como parte de outro trabalho de pesquisa que ampliou os recursos de visualização da EvoTrack (SILVA, 2010). Porém, esta funcionalidade ainda precisa ser incorporada ao módulo de visualização de redes sociais.

5.5. Apoio Computacional

Neste trabalho de pesquisa, foi concebida uma infraestrutura que integra a ferramenta COMPOOTIM ao Odyssey e à EvoTrack-SocialNetwork (Figura 5.16). Conforme visto no Capítulo 4, o ambiente Odyssey apoia a fase de definição ou criação da LPS através da modelagem das características de processo (com suas regras de composição) e da criação do modelo de contexto. Estas informações devem ser alimentadas na COMPOOTIM.

Nesta infraestrutura, a EvoTrack-SocialNetwork, abordada na seção anterior, e que contém os módulos de mineração, visualização e análise de redes sociais, tem o papel de alimentar a COMPOOTIM com dados reais da colaboração durante a execução do projeto de desenvolvimento. Isso permite o planejamento e acompanhamento do nível de colaboração do projeto.

Entretanto, vale ressaltar que apesar de ter sido originalmente concebida de forma integrada, ainda não foram criadas as interfaces entre os ambientes previstos nesta infraestrutura. Então, dados vindos do Odyssey, por exemplo, por enquanto são alimentados manualmente na COMPOOTIM. A EvoTrack-SocialNetwork ainda não possui integração com os demais ambientes.

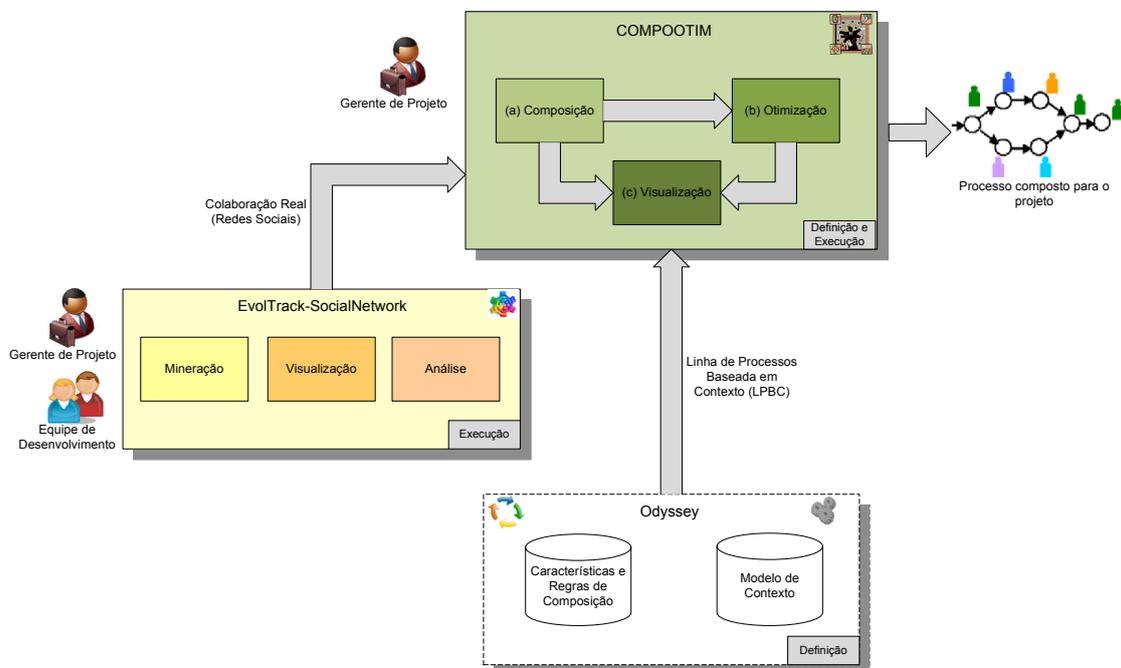


Figura 5.16 – Visão Geral da Infraestrutura Computacional

5.5.1. Funcionalidades do Mecanismo de Composição

A tela inicial permite o acesso do usuário aos três principais mecanismos da COMPOOTIM (Figura 5.17): o mecanismo de composição de processo, o mecanismo de otimização da colaboração nos processos e o mecanismo de visualização dos processos sugeridos. Os usuários dessa ferramenta são o Grupo de Definição de Processos de Software (GDPS) e o Gerente de Projeto.

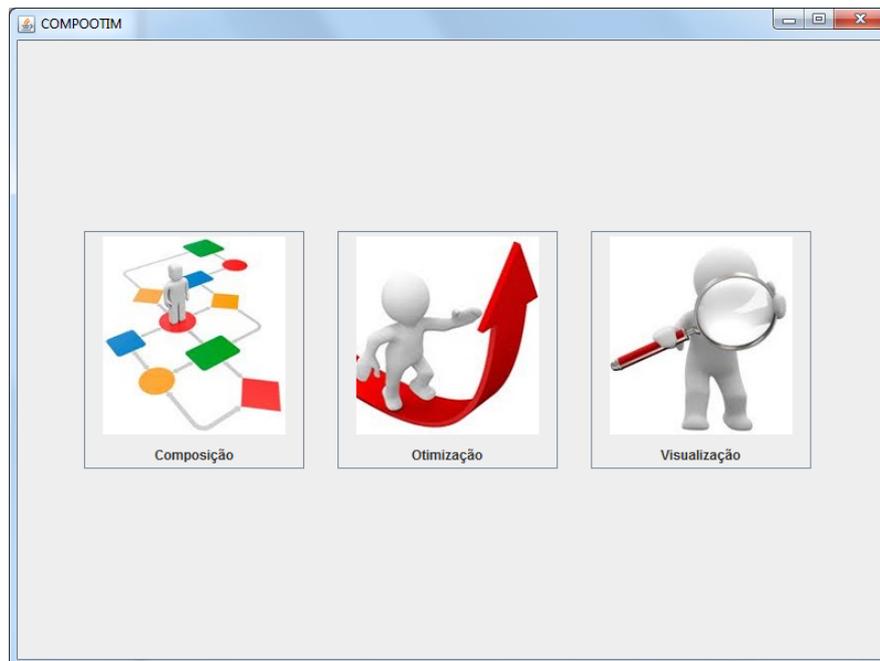


Figura 5.17 – Tela inicial da COMPOOTIM

Para a criação da LPS, a Figura 5.18 mostra um exemplo do cadastro das características de processo na COMPOOTIM. Cada característica possui um nome único, é classificada de acordo com o tipo (que segue a notação *Odyssey-ProcessFEX* vista no Capítulo 4), tem uma descrição textual e também é classificada quanto à opcionalidade e variabilidade. Além disso, podem ser definidos os relacionamentos (previstos na notação *Odyssey-ProcessFEX*) entre as características.

De forma similar, as regras de composição de características, dimensões e informações de contexto também são registradas na COMPOOTIM, concluindo assim a fase de análise de domínio de processos.

Característica	Tipo	Opcionalidade	Variabilidade
Acompanhamento	Atividade	Mandatório	Invariante
Acompanhamento de Projeto	Atividade	Mandatório	Invariante
Acompanhar Progresso da Iteração	Tarefa	Mandatório	PontoVariação
Alocar Tarefas	Tarefa	Opcional	PontoVariação
Atribuir Responsabilidades	Tarefa	Opcional	Invariante
Atribuir Tarefa	Tarefa	Mandatório	Invariante
Avaliar Iteração	Tarefa	Mandatório	PontoVariação
Avaliar Resultados do Projeto	Tarefa	Opcional	Invariante
Avaliação da Iteração	Produto	Opcional	Invariante
Concepção de Projeto	Atividade	Opcional	Invariante
Definir Equipe	Tarefa	Opcional	Invariante
Definir Equipe e Organização do Projeto	Tarefa	Opcional	Invariante
Definir e Priorizar Tarefas	Tarefa	Mandatório	Invariante
Desenvolver Plano da Iteração	Tarefa	Mandatório	Invariante
Desenvolver Plano de Aceitação de Produto	Tarefa	Opcional	Variante

Figura 5.18 – Características de processo na COMPOOTIM

Já na fase de projeto de domínio de processos devem ser registrados os componentes de processos (Figura 5.19) e as regras de composição dos componentes. Para cada componente devem ser informadas ainda as suas interfaces requeridas, interfaces providas e relacionamentos. Em seguida, o componente de processo pode ser associado com as práticas de colaboração. A COMPOOTIM já aplica os pesos de acordo com as práticas selecionadas e calcula o potencial de colaboração do componente, conforme definido na estratégia de medição da colaboração (Figura 5.20).

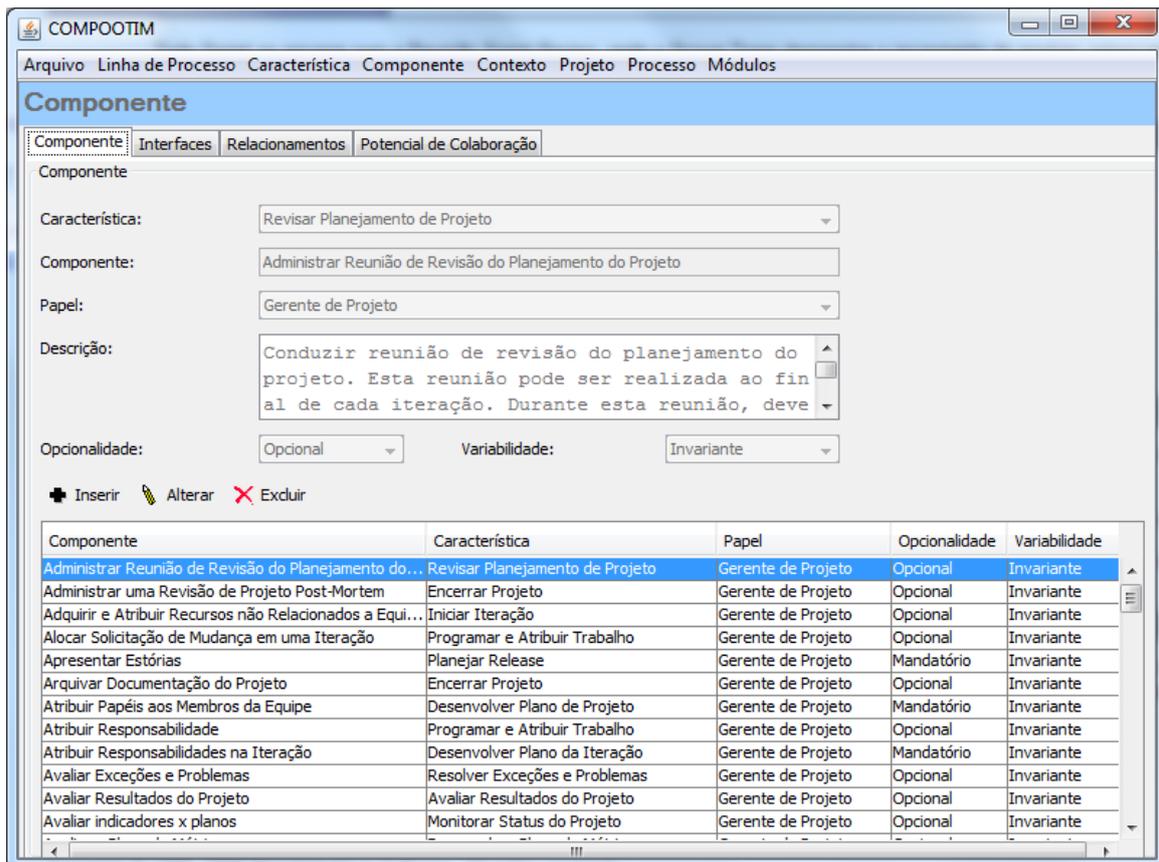


Figura 5.19 – Componentes de Processo na COMPOOTIM

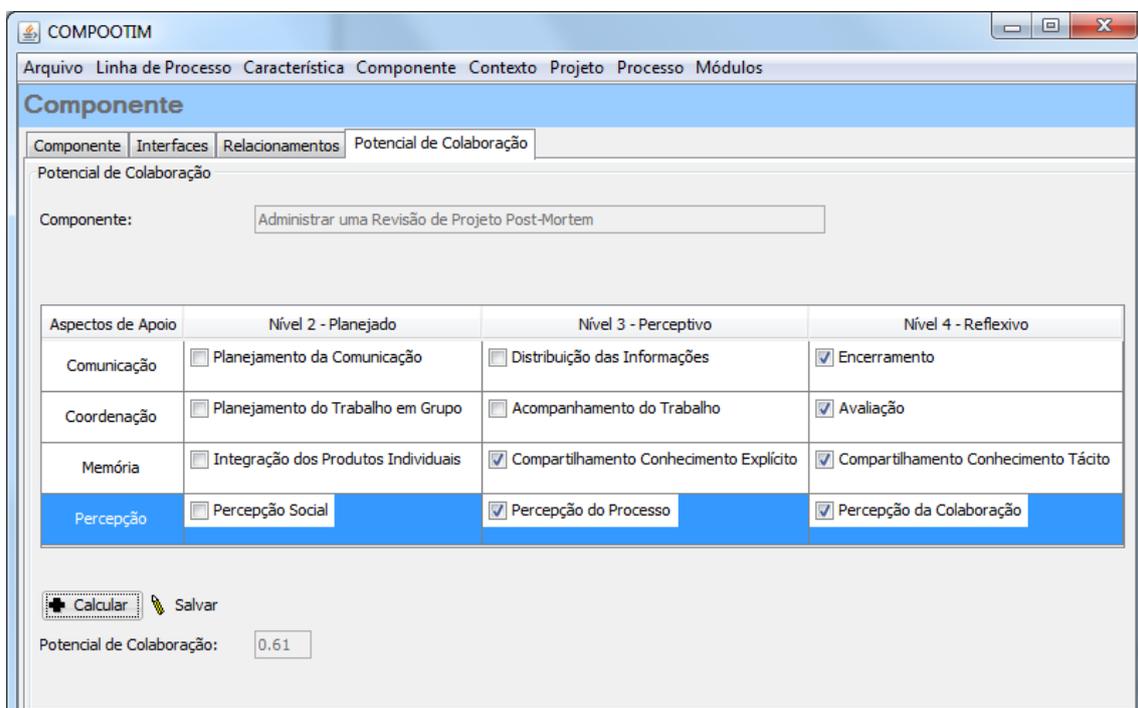


Figura 5.20 – Cálculo do Potencial de Colaboração do Componente na COMPOOTIM

Para concluir o modelo de contexto e a fase de projeto de domínio, também devem ser registradas as situações e regras de contexto. As regras de contexto seguem a estrutura da UbiFEX e são ilustradas na Figura 5.21.

Regra de Contexto	Tipo de Regra
RCTX1	Inclusiva
RCTX10	Inclusiva
RCTX11	Inclusiva

Figura 5.21 – Regras de Contexto na COMPOOTIM

Uma vez criada a LPS, inicia-se a fase de análise de aplicação de processos com a caracterização do projeto. Os valores das informações de contexto do projeto são apresentados na Figura 5.22.

Dimensão de Contexto	Informação de Contexto	Valor
Equipe	EstabilidadeEquipe	Baixa
Equipe	ExperienciaDesenvolvimentoSoftware	Baixa
Equipe	ExperienciaDominio	Baixa
Equipe	ExperienciaGerencial	MuitoAlta

Figura 5.22 – Informações de Contexto do Projeto na COMPOOTIM

Uma vez caracterizado o contexto do projeto, a COMPOOTIM é capaz de identificar as situações e regras de contexto que se aplicam ao projeto (Figura 5.23). Ela começa determinando as situações de contexto do projeto a partir da combinação das informações de contexto do projeto (Figura 5.22) e das situações de contexto pré-definidas.

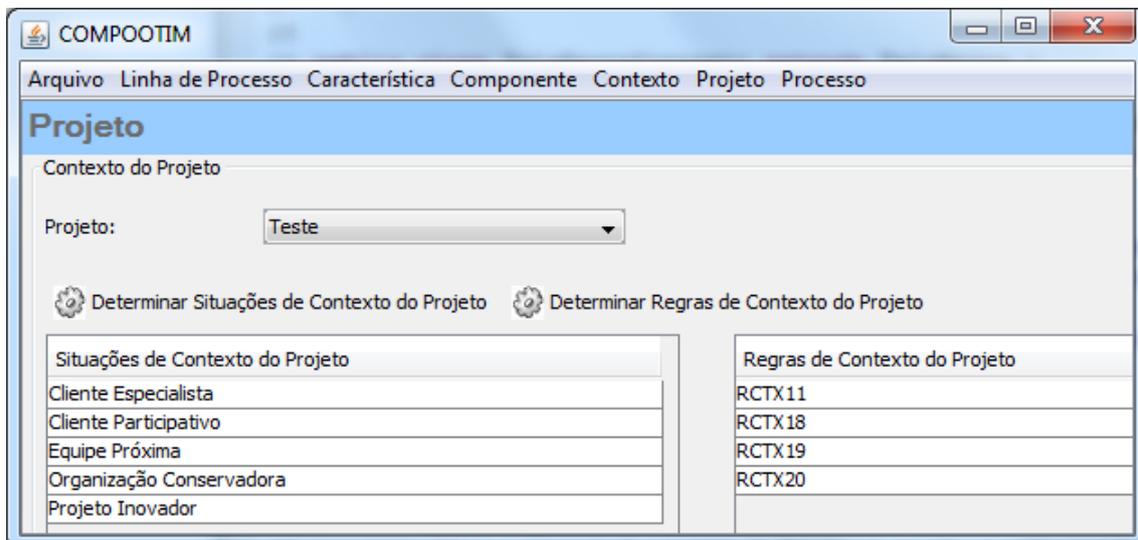


Figura 5.23 – Situações e Regras de Contexto do Projeto determinadas pela COMPOOTIM

Usando as regras de contexto aplicáveis ao projeto e as regras de composição de características definidas previamente, são inferidas as características do processo que devem estar presentes no processo resultante. Em seguida, as regras de composição de características são aplicadas e também incluem ou excluem algumas características (Figura 5.24). Neste momento do filtro, o Gerente de Projeto também pode tomar algumas decisões sobre configuração de variabilidades e opcionalidades e assinalar as características que ele deseja que sejam recortadas para o processo.

O filtro dos componentes seleciona apenas aqueles componentes associados às características que foram incluídas para o projeto. Em seguida, são aplicadas as regras de composição dos componentes que também incluem ou excluem alguns componentes do processo (Figura 5.25).

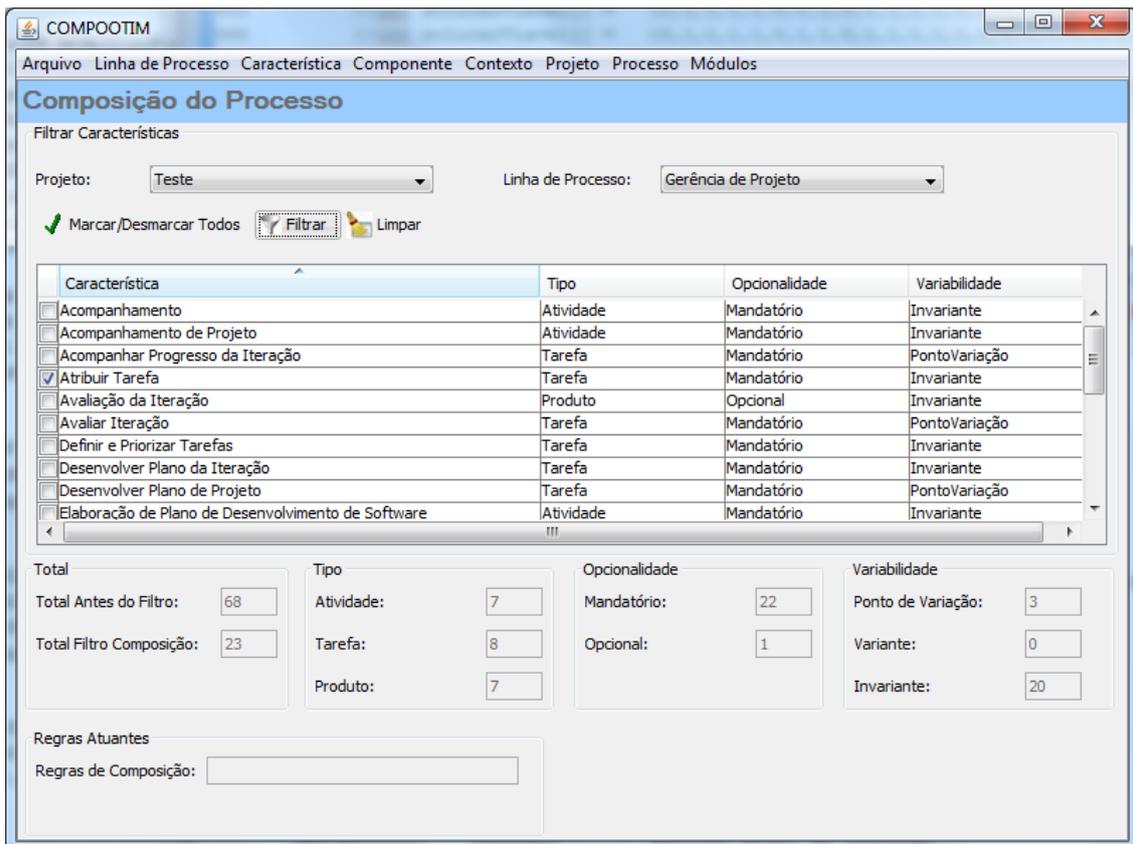


Figura 5.24 – Filtro das Características pelas Regras de Contexto e Composição aplicadas pela COMPOOTIM

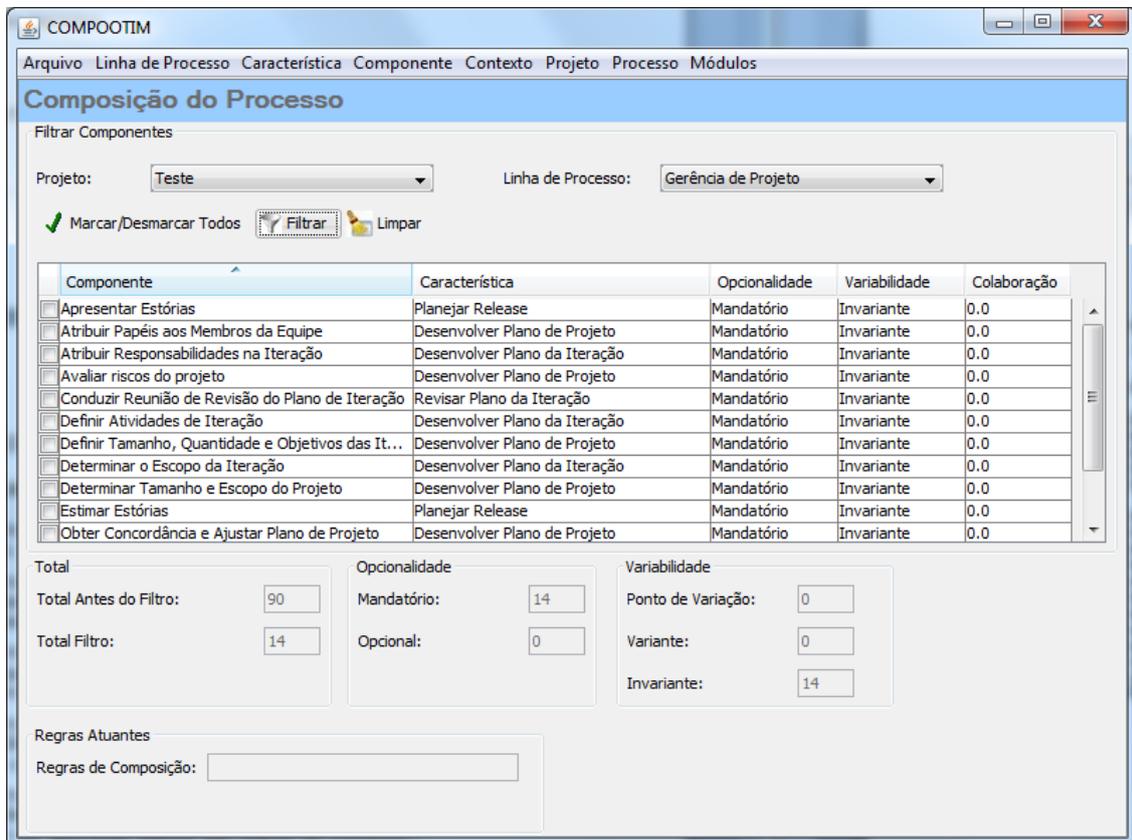


Figura 5.25 – Filtro dos Componentes pelas Características e Regras de Composição aplicadas pela COMPOOTIM

5.5.2. Funcionalidades do Mecanismo de Otimização

Nesta etapa, o Gerente de Projeto poderá escolher a estratégia de medição da colaboração que deseja que seja aplicada ao seu projeto. A partir dessa escolha, o mecanismo de otimização irá sequenciar os componentes e otimizar a colaboração, conforme descrito na Seção 5.3.2.

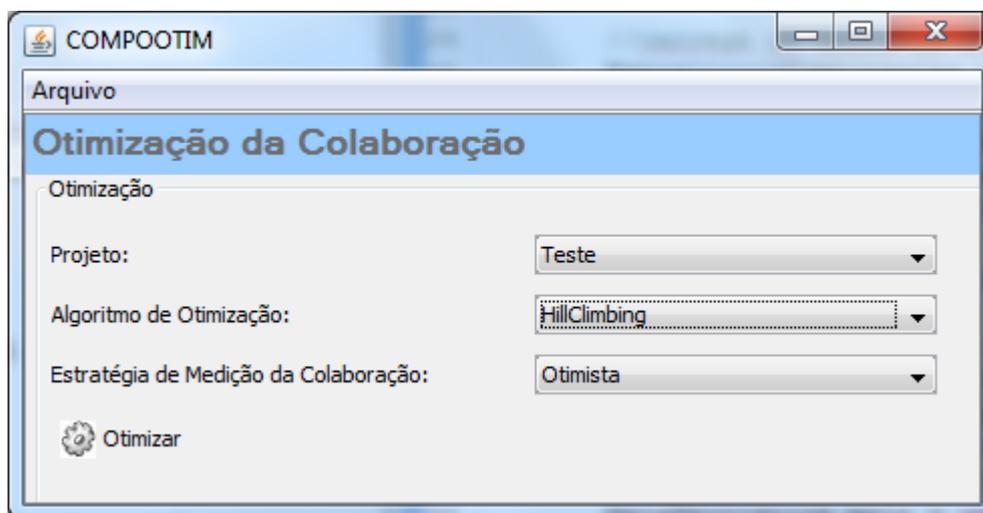


Figura 5.26 – Início da otimização da colaboração na COMPOOTIM

No mecanismo de otimização foram implementados três algoritmos – HC, RS e Força Bruta (FB). Os algoritmos foram implementados usando o *framework* de heurísticas JMetal (DURILLO et al., 2006), desenvolvido em Java com o objetivo de oferecer suporte à implementação, execução e estudo e técnicas de busca heurística. A implementação dos algoritmos segue a lógica descrita na modelagem do problema na Seção 5.3.2.

Os dois primeiros algoritmos trabalham com um limite máximo de avaliações correspondente ao número de componentes opcionais elevado ao quadrado. O FB é usado para avaliar se os dois primeiros são capazes de encontrar as soluções ótimas.

5.5.3. Funcionalidades do Mecanismo de Visualização

Após a execução da otimização, as opções de processos resultantes são apresentadas no mecanismo de visualização (Figura 5.27). Estas opções variam em relação ao potencial de colaboração alcançado, o número e a ordem de execução dos componentes e o número de artefatos que constituem o processo.

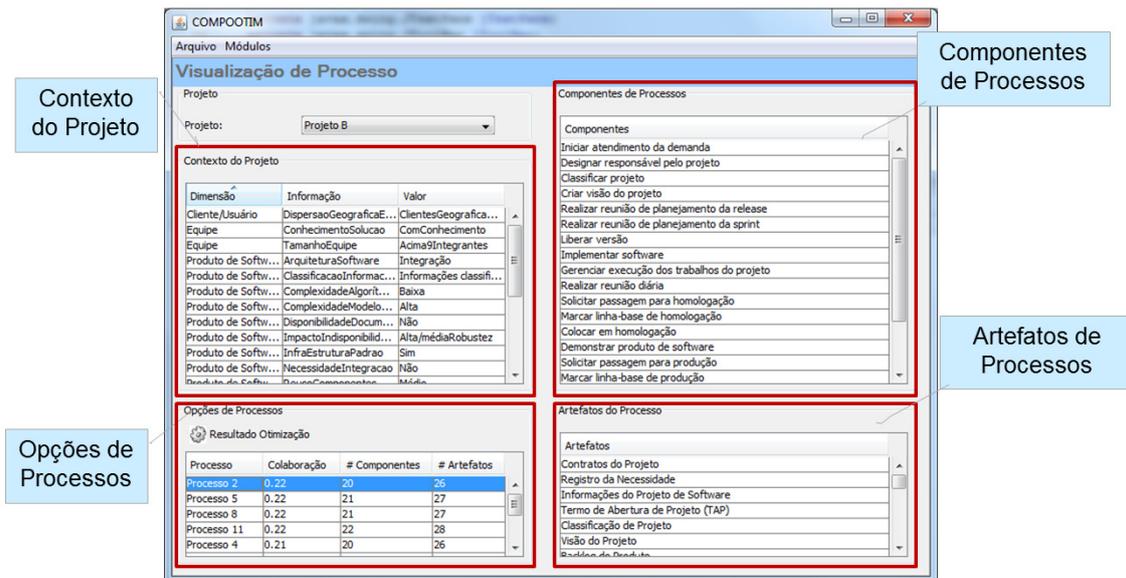


Figura 5.27 – Visualização das Opções de Processos na COMPOOTIM

5.6. Considerações Finais

Este capítulo apresentou a COMPOOTIM como uma solução composta por instrumentos para planejar, compor, otimizar e acompanhar a colaboração.

No planejamento, o CollabMM foi adotado para definir os objetivos de colaboração do processo associados aos seus níveis de maturidade em colaboração e suas características.

A estratégia de medição da colaboração apresentada permite definir o PC de cada componente de processo e, a partir da escolha da estratégia desejada, calcula a colaboração do processo como um todo. Este tema também será explorado por (CLARET, 2013) devido às múltiplas oportunidades de pesquisa futuras interessantes.

Em relação aos instrumentos de composição, a sistemática da COMPOOTIM define o passo-a-passo para a criação de uma LPS e para a composição dos processos a partir da LPS. O ferramental de apoio construído inclui mecanismos de composição, otimização e visualização.

O mecanismo para a otimização da colaboração proposto incluiu algoritmos de Força Bruta, *Random Search* e *Hill Climbing*.

A partir dos níveis de colaboração do CollabMM, o trabalho de Santos *et al.* (2010) mostrou ser possível combinar as propriedades de ARS para identificar características de colaboração e acompanhar o nível de colaboração

presente na execução de um processo de software. Este capítulo apresentou ainda a ferramenta de visualização e análise de redes sociais EvoTrack-SocialNetwork.

Também foi fornecida uma visão geral do apoio computacional proposto, com alguns detalhes técnicos sobre seu desenvolvimento e ilustradas as suas principais funcionalidades.

No próximo capítulo são descritos os estudos experimentais realizados para avaliar as soluções propostas nesta tese.

6. Estudos Experimentais

Este capítulo reúne os estudos experimentais realizados para avaliar os instrumentos propostos como parte da solução COMPOOTIM. Primeiramente, é resumido um estudo exploratório para avaliar o cálculo do potencial de colaboração dos componentes de processo. Em seguida, é apresentado um estudo exploratório sobre o acompanhamento da colaboração nas redes sociais de desenvolvimento de software, usando a EvolTrack-SocialNetwork. Por fim, é descrito o estudo de caso realizado em um contexto real da indústria para avaliação da sistemática e ferramental de apoio para composição de processos.

Com o objetivo de avaliar a solução da COMPOOTIM, apresentada no capítulo anterior, foram realizados alguns estudos experimentais no contexto deste trabalho de pesquisa. Seguindo a abordagem de avaliação incremental proposta por Shull *et al.* (2001), foram realizados inicialmente alguns estudos exploratórios, envolvendo a estratégia de medição da colaboração e a ferramenta de visualização e análise de redes sociais EvolTrack-SocialNetwork, antes de ser realizado um estudo de caso em um ambiente real da indústria, com uma empresa de petróleo de grande porte no Rio de Janeiro.

Neste estudo de caso pretendia-se avaliar a aplicabilidade, viabilidade e facilidade de uso da sistemática. O estudo de caso foi realizado em duas etapas. Na primeira etapa, o objetivo era avaliar se a sistemática da COMPOOTIM é capaz de gerar uma Linha de Processos de Software (LPS). Assim, como resultado desta etapa, foi criada e avaliada uma LPS que serviu para termos uma base de dados confiável populada para continuidade do estudo de caso. Na segunda etapa, o objetivo era avaliar se a COMPOOTIM era capaz de, a partir da LPS criada na etapa anterior, compor processos para dois projetos de desenvolvimento escolhidos, de acordo com o contexto desses projetos e de forma otimizada, usando o ferramental de apoio construído.

Além disso, para avaliar os processos sugeridos do ponto de vista da otimização da colaboração, também foi utilizada análise estatística para comparar os resultados obtidos com os diferentes algoritmos de otimização em relação ao *fitness* e ao tempo de execução.

O restante do capítulo está organizado da seguinte forma: a Seção 6.1 resume um estudo exploratório para avaliar o cálculo do potencial de colaboração dos componentes de processo; a Seção 6.2 descreve um estudo exploratório realizado para avaliar se é possível acompanhar a colaboração

entre os membros da equipe de desenvolvimento, usando a ferramenta EvolTrack-SocialNetwork; a Seção 6.3 descreve o planejamento, a execução, os resultados e as conclusões da primeira etapa do estudo de caso; a Seção 6.4 segue a mesma estrutura, mas no escopo da segunda etapa do estudo de caso; a Seção 6.5 discute a avaliação da otimização da colaboração realizada na composição dos processos. Por fim, a Seção 6.6 conclui o capítulo.

6.1. Estudo Exploratório – Medição da Colaboração

Como uma primeira iniciativa de avaliar a estratégia de medição da colaboração proposta no Capítulo 5, foi realizado um estudo exploratório²⁰ com 4 especialistas em colaboração. Usando os 10 componentes de processo apresentados na Tabela 6.1, os participantes do estudo indicaram as práticas de colaboração associadas a cada um dos componentes. Assim, foi calculado o PC atribuído por cada participante. O PC resultante para cada componente de processo na Tabela 5.5 é resultado da mediana ponderada dos PCs indicados por cada participante.

Tabela 6.1 – Descrição dos Componentes de Processo

CP6 – Constituir Equipe para Execução do Projeto
Negociar, junto ao(s) líder(es) de equipe(s), a alocação de um ou mais profissionais para atuar na execução do projeto e definir o prazo de alocação do recurso e o perfil necessário.
CP7 – Elaborar Cronograma do Projeto
Elaborar o cronograma do projeto, tendo como base a Estrutura Analítica do Projeto (EAP), definindo atividades, sequenciando-as, estimando recursos e duração para a realização das mesmas.
CP11 – Elaborar Plano de Comunicação
Realizar o planejamento (tipo, sequência, forma e etc.) da distribuição de informações do projeto para os principais interessados do projeto.
CP19 – Distribuir Informações do Projeto
Distribuir as informações do projeto aos principais interessados. A distribuição das informações pode acontecer em função de solicitações esporádicas (não planejadas) ou em conformidade com o Plano de Comunicação (planejadas).
CP20 – Acompanhar Desempenho do Projeto
Monitorar e controlar o desempenho do projeto, identificando eventuais desvios em escopo, prazo e custo, suas causas, verificar o tratamento dado a estes desvios e promover as ações corretivas ou preventivas necessárias.
CP35 – Realizar Reunião Diária
Tem como objetivo melhorar a comunicação, identificar e remover impedimentos e riscos para o desenvolvimento, ressaltar e promover a tomada rápida de decisões e

²⁰ O plano e os instrumentos do estudo estão disponíveis em: <http://reuse.cos.ufrj.br/cdsoft/compoitim-estudo-potcolab.html>

CP6 – Constituir Equipe para Execução do Projeto
<p>aumentar o nível de conhecimento de todos acerca do projeto. É fortemente recomendado que os membros do time se encontrem diariamente para uma reunião de aproximadamente 15 minutos.</p>
CP40 – Detalhar Casos de Uso
<p>Detalhar os casos de uso identificados para o sistema mostrados no diagrama de casos de uso, através da construção do fluxo básico e fluxos alternativos.</p>
CP61 – Executar Testes
<p>O testador deve executar os testes e esta execução deve ser evidenciada na ferramenta de teste ou Documento de Caso de Teste. Ao final da execução de um determinado teste, os defeitos encontrados poderão ser reportados imediatamente à equipe do projeto, conforme avaliação e necessidade do testador, para solicitar ações imediatas de correção antes da finalização do ciclo de teste.</p>
CP26 – Registrar Lições Aprendidas
<p>Registrar as lições aprendidas observadas durante o projeto. O responsável pelo projeto, em consenso com a equipe do projeto, define a periodicidade da ocorrência desta atividade.</p>
CP29 – Encerrar Projeto
<p>O encerramento deverá ser formalizado através do Relatório de Fechamento de Projeto.</p>

Tabela 6.2 – Exemplos de Componentes de Processo e seus respectivos PCs

Componente	P1	P2	P3	P4	Resultado
CP1	2,00	2,00	2,00	3,00	2,00
CP2	2,00	2,50	2,00	3,00	2,25
CP3	2,00	2,00	2,00	3,00	2,00
CP4	3,00	3,00	3,00	4,00	3,00
CP5	3,50	3,00	4,00	4,00	3,75
CP6	2,00	0,00	2,00	3,00	2,00
CP7	3,00	3,00	3,00	4,00	3,00
CP8	4,00	3,00	4,00	4,00	4,00
CP9	4,00	4,00	4,00	4,00	4,00
CP10	4,00	4,00	4,00	4,00	4,00

Vale ressaltar que durante este estudo exploratório, foram feitas algumas observações interessantes pelos participantes que ressaltam algumas limitações da proposta e/ou indicam oportunidades de pesquisa futuras:

- **Alguns componentes de processo contribuem mais para uma prática do que para outra** – Um dos participantes destacou a dificuldade de associar as práticas aos componentes de processo dando a todas elas a mesma relevância, pois um componente de processo pode contribuir muito pouco para uma determinada prática e muito para outra e na estratégia de medição proposta para o cálculo do PC dos componentes, não são atribuídos pesos distintos de acordo com o grau de contribuição;

- **Resultado da execução do processo** – Potencialmente, um componente pode contribuir para várias práticas de colaboração, mas durante sua execução isso pode não acontecer. A estratégia de medição do PC do componente, de fato, é uma medida atribuída durante o planejamento da colaboração. Porém, este PC pode não se confirmar durante a execução do componente de processo. Por exemplo, o “Registrar Lições Aprendidas” pode levar a diversas percepções, mas não necessariamente. Então, durante o acompanhamento da colaboração, existe a necessidade de avaliar ao longo do tempo (com base na execução do processo) os valores assumidos para cada componente.

6.2. Estudo Exploratório – EvoTrack-SocialNetwork

Um estudo exploratório foi planejado e conduzido neste trabalho, no contexto de projetos de software livre, para avaliar se é possível acompanhar a colaboração entre os membros da equipe de desenvolvimento usando a ferramenta EvoTrack-SocialNetwork (MAGDALENO *et al.*, 2012b).

6.2.1. Planejamento do Estudo Exploratório

Esta seção descreve o planejamento do estudo exploratório que seguiu o modelo definido em Travassos *et al.* (2002). O plano completo do estudo pode ser encontrado em (EVOLTRACK-SOCIALNETWORK, 2013).

6.2.1.1. Objetivo e Escopo

Este estudo exploratório tem o seguinte objetivo (apresentado de acordo com o paradigma GQM (*Goal Question Metric*) (BASILI *et al.*, 1994)) e escopo:

Analisar a colaboração nas redes sociais de desenvolvimento de software usando a ferramenta de visualização e análise de redes sociais EvoTrack-SocialNetwork

Com o propósito de caracterizar

Com respeito à efetividade

Do ponto de vista do pesquisador da área de colaboração

No contexto de projetos de desenvolvimento de software livre

6.2.1.2. Contexto

O contexto descreve as condições em que o estudo será executado (TRAVASSOS *et al.*, 2002).

a) Participantes

A escolha dos participantes foi baseada em princípios não probabilísticos e a população foi determinada por conveniência. Os participantes do estudo são alunos e ex-alunos de pós-graduação da Universidade Federal do Estado do Rio de Janeiro (UNIRIO). Como se trata de um estudo exploratório, foram selecionados como participantes pesquisadores da área de colaboração para capturar a visão de especialistas desta área.

As sessões do estudo foram individuais. Todos os participantes utilizaram a ferramenta. Não houve nenhum tipo de compensação para os participantes.

b) Tarefas

Neste estudo foi definido um conjunto de tarefas a serem realizadas, visando caracterizar o apoio à percepção da colaboração. As tarefas propostas (Tabela 6.3) exploram tanto a visualização quanto a ARS. As tarefas são divididas em três grupos, com diferentes níveis de complexidade, baseados no trabalho de Oliveira (2011): (i) As tarefas mais fundamentais e simples foram classificadas como *tarefas de filtragem*; (ii) *tarefas básicas* são as que podem ser resolvidas por meio de fatos extraídos da visualização ou análise; (iii) *tarefas de assimilação* exigem dos participantes que usem seu conhecimento e raciocínio para interpretar a informação.

Tabela 6.3 – Tarefas propostas para o estudo

Id	Dificuldade	Questão
Q1	Filtragem	Quais atores compõem a versão 1014 da rede social?
Q2	Filtragem	Qual é a densidade da versão 1036 da rede sócio-técnica?
Q3	Básica	Qual é a distância entre a classe TedMainDialog e a classe TedMainToolBarButton na versão 1036 da rede técnica?
Q4	Básica	Quem tem a maior centralidade de grau na versão 1036 da rede sócio-técnica?
Q5	Básica	O que mudou na rede social da versão 900 para a versão 1014?
Q6	Básica	Com quem mhstead pode estar em conflito na versão 900?
Q7	Assimilação	Qual artefato tem indícios de que precisa ser refatorado?
Q8	Assimilação	Para quem você passaria a tarefa de trabalhar no artefato GeneralPanel?
Q9	Assimilação	Com quem ted_jofo precisa se comunicar para coordenar o

Id	Dificuldade	Questão
		trabalho?
Q10	Assimilação	De acordo com as características abaixo (Capítulo 5), como você classificaria a coordenação do projeto TED?

c) Dados Utilizados

Este estudo exploratório utilizou dados de projetos de desenvolvimento de software livre. A escolha do ambiente de projetos de software livre para a realização dos estudos iniciais se deve ao fato de que estes projetos publicam os seus processos de desenvolvimento, códigos, artefatos e dados sobre os produtos livremente na Internet. Portanto, fornecem uma oportunidade única para as experimentações em ES, devido à diversidade, complexidade e representatividade de dados disponíveis em uma escala global.

Este estudo coloca o foco em redes sociais obtidas a partir da mineração das relações existentes implicitamente entre os desenvolvedores de software por meio das atualizações de código fonte realizadas no repositório de gerência de configuração. Assume-se que estas relações podem ser uma fonte de informação adequada para entender a coordenação do grupo (SANTOS *et al.*, 2011). A partir do código-fonte, é possível verificar quais atividades foram realizadas pelos membros do grupo e verificar o andamento do projeto.

Em particular, este estudo utilizou o projeto de software livre Torrent Episode Downloader (TED)²¹ extraído do portal SourceForge. O projeto foi selecionado randomicamente, levando-se em conta o fato de ter sido desenvolvido em Java para satisfazer às restrições tecnológicas da ferramenta. Também foi selecionado um projeto com pelo menos três desenvolvedores, para ter um mínimo de riqueza de atores e relacionamentos a serem explorados. Os dados minerados desse projeto resultaram em mais de 120 artefatos e 4 desenvolvedores.

6.2.1.3. Instrumentação e Preparação

Para este estudo, foram projetados e revisados (por outros pesquisadores) cinco instrumentos principais. O primeiro instrumento é o termo de consentimento livre e esclarecido (Apêndice I), que declara ao participante o objetivo do estudo, os limites da sua participação e suas responsabilidades

²¹ Site TED: <http://sourceforge.net/projects/ted/>

durante a avaliação. Este termo foi assinado pelo participante antes do início do estudo.

Em seguida, o participante foi solicitado a preencher um formulário de caracterização (Apêndice II), que permite ao pesquisador determinar o seu perfil e auxilia na análise posterior dos dados obtidos por meio do estudo.

Ainda antes de iniciar o estudo, foi ministrado um breve treinamento com os participantes visando explicar os principais conceitos de redes sociais e a utilização da ferramenta. Este material encontra-se disponível no site da EvolTrack-SocialNetwork (EVOLTRACK-SOCIALNETWORK, 2013).

De forma a facilitar a imersão do participante no contexto do estudo, foi criada uma situação fictícia, na qual o participante é colocado na posição de gerente de um projeto de desenvolvimento de software cujo objetivo é criar um produto inovador. Assim, o formulário para a realização do estudo (Apêndice III) apresenta ao participante esse contexto de trabalho e as instruções para a realização das tarefas. Este formulário vem acompanhado da apresentação dos dados na ferramenta. Este conjunto de informações deve ser utilizado pelo participante ao longo de toda a execução do estudo. Por fim, este formulário também é o instrumento para a coleta dos resultados da tarefa.

Finalmente, após a execução da tarefa, foi entregue ao participante um questionário para avaliação do estudo (Apêndice IV), que pretende obter informações qualitativas acerca do estudo, do ponto de vista do participante. Este questionário inclui sugestões para a ferramenta EvolTrack-SocialNetwork e considerações sobre o procedimento de avaliação.

6.2.1.4. Validade do Estudo

É comum que haja questões que possam impactar ou limitar a validade dos resultados dos estudos. Estas questões são denominadas ameaças a validade (TRAVASSOS *et al.*, 2002, WOHLIN *et al.*, 1999). As ameaças a validade identificadas para o presente estudo são:

a) Validade interna

- Caso o estudo não seja executado em um único dia por todos os participantes, um ponto que poderia influenciar o resultado do estudo seria a troca de informações entre os participantes que já haviam realizado o estudo e os que ainda não o haviam realizado. Para evitar este problema,

foi requisitado explicitamente que os participantes não trocassem informações a respeito do estudo;

- Os participantes do estudo são colegas de trabalho e membros do mesmo grupo de pesquisa do pesquisador. Dessa forma, pode haver uma tentativa de agradar o pesquisador. Para minimizar esse risco, tanto nos instrumentos utilizados como no treinamento realizado no início do estudo, buscou-se deixar claro que os participantes devem ser imparciais na sua análise e que os resultados do estudo não iriam influenciar o resultado do desempenho dos participantes nas suas atividades. No entanto, não é possível garantir se os participantes executaram o estudo da mesma forma que o teriam executado em outro cenário;
- A própria ferramenta de apoio pode influenciar os resultados, caso os participantes enfrentem dificuldades não esperadas na realização do estudo (lentidão, erros no servidor, etc.). A interação com a ferramenta também pode influenciar na maneira como os participantes realizam a atividade. Assim, foi ministrado um treinamento sobre a utilização da ferramenta e durante o estudo foi observada a interação com a ferramenta para avaliar se algum fator não planejado em relação à ferramenta influenciou na execução das atividades;
- O entendimento dos participantes sobre as questões dos formulários é diretamente influenciado pela forma como as questões foram elaboradas: se a questão tiver sido mal formulada, o estudo pode ser afetado negativamente (WOHLIN *et al.*, 1999). A análise dos instrumentos utilizados pelos outros pesquisadores envolvidos no estudo visa justamente reduzir esta interferência.

b) Validade externa

- Normalmente, ambientes acadêmicos não simulam totalmente as condições existentes em um ambiente de desenvolvimento de software. Em contrapartida, Carver *et al.* (2003) apontam que a realização de estudos experimentais tendo alunos como participantes permite obter uma série de benefícios para os pesquisadores (e.g., obtenção de evidências preliminares para confirmar ou rejeitar hipóteses e treinamento de jovens investigadores no campo da pesquisa empírica) e os estudantes (e.g.,

atividades práticas e a educação em tópicos que são alvo de pesquisas recentes);

- O estudo considera um único projeto de software livre. Assim, experimentos com uma quantidade maior ou com outros tipos de projetos devem ser executados;
- Não é possível representar todas as situações possíveis de um projeto de software. Apesar das tarefas do estudo terem sido elaboradas com base em situações comuns no contexto em que estão inseridas, é necessário verificar se os objetivos da EvolTrack-SocialNetwork são atingidos em outras circunstâncias.

c) Validade de construto

- O agrupamento das tarefas por tipo auxilia a análise dos dados. No entanto, embora algumas destas tarefas possam ter grau de dificuldade maior do que o de outras tarefas, o mesmo peso foi atribuído a todas as tarefas. Isto pode influenciar os resultados. Devido à subjetividade na avaliação do grau de dificuldade (o que introduziria viés na análise dos dados), optou-se por manter esta configuração.

d) Validade de conclusão

- O tamanho da amostra, com um número pequeno de participantes, não é o ideal do ponto de vista estatístico. Por isso, este estudo apresenta uma limitação nos resultados que serão considerados apenas como indícios. Além disso, será realizada apenas uma análise qualitativa dos resultados.

6.2.2. Execução do Estudo Exploratório

O estudo exploratório (MAGDALENO *et al.*, 2012b) foi realizado em 2011, com quatro participantes, em sessões individuais, durante uma semana. Os participantes foram dois alunos de mestrado e dois mestres (Tabela 6.4) na área de colaboração, do mesmo grupo de pesquisa, que não tiveram contato anterior com a EvolTrack-SocialNetwork. O perfil dos participantes (Tabela 6.4) indica um alto nível de experiência com ES e colaboração, mas eles não são especialistas em Gerenciamento de Projetos. Além disso, os participantes afirmaram, na média, baixa familiaridade com projetos de software livre ou com outras ferramentas de redes sociais. Em particular, as ferramentas Ariadne, Pajek e SVNAT foram mencionadas.

Tabela 6.4 – Caracterização dos Participantes

Participante	Formação Acadêmica	Grau de Experiência		
		Engenharia de Software	Gerência de Projetos	Colaboração
P1	Mestrado em andamento	3	0	4
P2	Mestrado concluído	2	2	3
P3	Mestrado em andamento	4	2	3
P4	Mestrado concluído	4	3	3

Legenda: 0 = nenhum; 1 = conhecimento teórico; 2 = conhecimento teórico aplicado ao contexto acadêmico; 3 = conhecimento teórico com experiência individual; 4 = experiência em projetos reais

6.2.3. Análise dos Resultados do Estudo Exploratório

Esta seção resume os principais resultados obtidos com a execução deste estudo exploratório. Os resultados foram estruturados de acordo com as variáveis avaliadas.

6.2.3.1. Corretude

Durante o estudo, os participantes usaram a EvoTrack-SocialNetwork para responder às dez questões propostas. A Tabela 6.5 resume os resultados das respostas às tarefas do estudo. Na média, os participantes responderam corretamente 8,25 questões.

Tabela 6.5 – Resultados das respostas às tarefas do estudo

Id	Questão	P1	P2	P3	P4
Q1	Quais atores compõem a versão 1014 da rede social?	✓	✓	✓	✓
Q2	Qual é a densidade da versão 1036 da rede sócio-técnica?	✓	✓	✗	✗
Q3	Qual é a distância entre a classe TedMainDialog e a classe TedMainToolBarButton na versão 1036 da rede técnica?	✗	✗	✗	✓
Q4	Quem tem a maior centralidade de grau na versão 1036 da rede sócio-técnica?	✗	✓	✓	✓
Q5	O que mudou na rede social da versão 900 para a versão 1014?	✓	✓	✓	✓
Q6	Com quem mhstead pode estar em conflito na versão 900?	✓	✓	✓	✓
Q7	Qual artefato tem indícios de que precisa ser refatorado?	✓	✓	✓	✓
Q8	Para quem você passaria a tarefa de trabalhar no artefato GeneralPanel?	✓	✓	✓	✓
Q9	Com quem ted_jofo precisa se comunicar para coordenar o trabalho?	✓	✓	✓	✓
Q10	De acordo com as características abaixo, como você classificaria a coordenação do projeto TED?	✓	✗	✓	✓

A questão com a maior quantidade de erros foi a Q3, pois existiam classes com nomes similares e eles responderam à questão com base na

primeira classe encontrada, sem conferir corretamente o seu nome. Em geral, os erros se concentraram nas questões (Q2, Q3, Q4 e Q10) relacionadas com as propriedades de ARS, o que pode indicar uma dificuldade dos participantes em consultar estas propriedades no painel de análise. Ao mesmo tempo, essa observação reforça a ideia de que a visualização e a análise de redes sociais precisam ser combinadas.

Apesar de apenas um dos participantes ter respondido incorretamente a Q10, todos eles tiveram dificuldade de classificar a rede de acordo com as características de coordenação. Na rede social usada (Figura 6.1), todos os atores estão conectados, o que caracteriza uma coordenação distribuída com máxima densidade da rede. Portanto, era esperado um resultado melhor nesta última questão. Porém, destacou-se a complexidade destas características, o que indicou um novo requisito: classificar automaticamente a rede. Este requisito não foi implementado na versão atual da ferramenta, mas é prioritário no desenvolvimento das próximas versões.

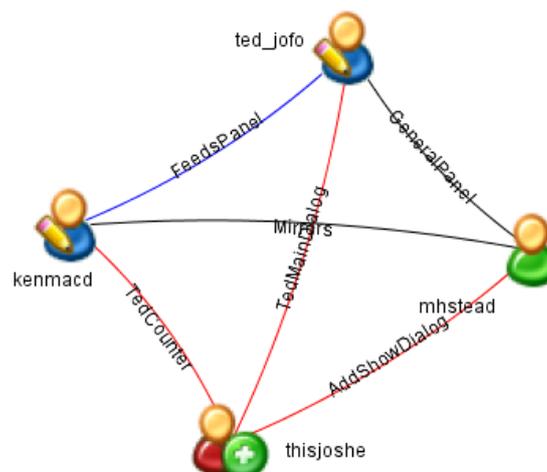


Figura 6.1 – Rede social de dependência do projeto TED

6.2.3.2. Tempo

A eficiência de tempo está resumida na Tabela 6.6. Em média, os participantes gastaram 22,75 minutos para responder às perguntas. O participante (P4), com a maior experiência, foi o que levou mais tempo para completar as tarefas. A média de tempo por tipo indica que a distribuição das perguntas estava equilibrada e que o nível de dificuldade realmente foi aumentando gradualmente.

Tabela 6.6 – Eficiência de tempo (em minutos)

Dificuldade	P1	P2	P3	P4	Média
Filtragem	2	2	1	1	1,5
Básica	8	6	5	7	6,5
Assimilação	15	15	9	20	14,75
Total	25	23	15	28	22,75

6.2.3.3. Satisfação com Resultados e Percepção da Colaboração

Ao final do estudo, os participantes preencheram o questionário e deram *feedback* qualitativo (Tabela 6.7) sobre o estudo. Em geral, consideraram-se capazes de executar as tarefas (Tabela 6.7a). Eles ficaram satisfeitos com os resultados obtidos, mas P3 enfrentou dificuldades para indicar a refatoração de um determinado artefato sem consultar o seu código fonte (Tabela 6.7b).

Tabela 6.7 – Feedback dos participantes

Item	P1	P2	P3	P4
a) Execução	Parcialmente	Sim	Sim	Sim
b) Satisfação	Sim	Sim	Parcialmente	Sim
c) Percepção	Sim	Sim	Sim	Sim
d) Contribuição	Facilitou bastante	Facilitou bastante	Facilitou bastante	Facilitou bastante
e) Dificuldade	Fácil	Muito fácil	Fácil	Fácil

Todos os participantes concordaram que é possível perceber como a colaboração acontece em um projeto usando as informações apresentadas (Tabela 6.7c).

6.2.3.4. Ferramenta

Os participantes também indicaram que a ferramenta contribuiu, facilitando a realização das tarefas propostas (Tabela 6.7d). As funcionalidades citadas como mais úteis foram: a visualização dos tipos de redes, as métricas existentes no painel de análise, a ampliação do tamanho dos nós e o uso de diferentes cores para assinalar as mudanças entre as versões das redes.

Os participantes avaliaram a ferramenta como “fácil” ou “muito fácil” de usar (Tabela 6.7e). A principal dificuldade mencionada foi a definição do melhor tipo de rede (técnica, sócio-técnica ou social) a ser utilizada em cada tipo de análise.

Perguntados sobre os aspectos positivos da ferramenta, os participantes forneceram as seguintes respostas. P1 respondeu: “*Visualização da rede, das interações, dos conflitos. Facilita a análise. Minerando na mão os dados seria difícil realizar as análises e muito demorado*”. P2 colocou: “*Análise das redes*”.

existentes é facilitada. A visualização dos relacionamentos existentes é evidenciada". Por sua vez, P3 afirmou: *"Sem a visualização, ficaria muito difícil perceber a colaboração nos projetos. E sem a busca e ordenação, seria muito difícil encontrar um elemento específico"*.

Em relação aos aspectos negativos da ferramenta, mencionou-se que a visualização pode ser difícil em grandes redes e que os filtros disponíveis na ferramenta não foram muito úteis, pois não são fáceis de utilizar e apresentaram alguns *bugs*. Os participantes também contribuíram com uma sugestão de melhoria em relação à possibilidade de migração de plataforma de Eclipse para Web.

Nos comentários adicionais, P3 destacou: *"Adorei o sistema. Muito útil para perceber a colaboração. Acredito ser útil também para desenvolvedores perceberem com quem estão trabalhando"*. P4 contribuiu espontaneamente com a seguinte afirmação: *"Achei a ferramenta ótima, muito intuitiva com diversos aspectos de percepção e colaboração que não foram explorados, até onde vai meu conhecimento, em nenhuma outra ferramenta de análise de redes sociais"*.

6.2.3.5. Melhorias da ferramenta

Consolidando as melhorias identificadas para a ferramenta é possível citar:

- Classificação automática da rede de acordo com as características de colaboração apresentadas no Capítulo 5. Com base nas propriedades de ARS calculadas, a ferramenta já poderia indicar no painel de análise a característica de colaboração mais representativa para cada versão da rede apresentada;
- Os filtros da ferramenta não foram particularmente úteis durante a realização do estudo, pois apresentaram alguns *bugs* e não ajudam a posicionar no grafo o nó ou relacionamento pesquisado;
- Também foram observadas oportunidades de melhorias relacionadas à escalabilidade de apresentação dos diagramas das redes. De fato, sabe-se que as ferramentas com visualizações gráficas podem ser ineficazes em projetos de grande dimensão, devido à sobreposição dos nós e arestas (COSTA *et al.*, 2011, SOUZA *et al.*, 2012). Recursos de *clustering* e *drill-down* estão sendo considerados para futuras versões. Estas

funcionalidades já foram implementadas como parte de outro trabalho de pesquisa do grupo e ampliou os recursos de visualização da EvoTrack (SILVA, 2010), mas ainda precisam ser incorporadas à EvoTrack-SocialNetwork.

6.2.3.6. Limitações

Ainda que este estudo exploratório tenha obtido resultados positivos, trata-se apenas de uma evidência inicial, devido ao número limitado de participantes neste estudo. Assim, estudos com uma quantidade maior ou outros tipos de projetos devem ser executados. Outra oportunidade é mostrar as redes para a comunidade do projeto de software livre escolhido e levá-los a refletir e comentar sobre a sua precisão. Desta forma, é possível complementar os resultados, obtidos através de mineração de dados e geração de redes, com perspectivas adquiridas via observação ou entrevistas com quem está de fato desenvolvendo o projeto para fornecer uma interpretação mais profunda.

Estas e outras questões abrem espaço para o planejamento de novas avaliações da proposta. Segundo a metodologia incremental proposta por Shull *et al.* (2001), este estudo exploratório é parte de um plano mais amplo de avaliação. Após a implementação de algumas das ideias reunidas a partir deste estudo, o passo seguinte na nossa agenda de pesquisa em relação à avaliação da EvoTrack-SocialNetwork seria a condução de um estudo de viabilidade com mais participantes (gerentes de projeto provenientes da indústria) e a utilização de dois projetos de software livre. O objetivo é avaliar o apoio da EvoTrack-Socialnetwork na tomada de decisão dos gerentes de projeto.

6.3. Estudo de Caso – Etapa 1 – Avaliação da Criação da Linha de Processos

Esta seção descreve a etapa de criação e avaliação da LPS a partir dos processos já existentes da organização.

6.3.1. Planejamento

Esta seção descreve o planejamento²² da Etapa 1 de avaliação da sistemática da COMPOOTIM. O propósito principal dessa etapa do estudo é avaliar se a

²² O plano completo da etapa 1 do estudo de caso pode ser obtido em: <http://reuse.cos.ufrj.br/cdsoft/compootim-estudo-criacaolinha.html>

sistemática da COMPOOTIM é capaz de apoiar a criação de uma LPS. Este objetivo pode ser detalhado de acordo com a estrutura proposta por (WOHLIN *et al.*, 1999):

Analisar a sistemática da COMPOOTIM

Com o propósito de caracterizar

Com respeito à viabilidade de criação da LPS

Do ponto de vista do gestor do Grupo de Definição de Processos de Software (GDPS)

No contexto de um ambiente real da indústria

6.3.1.1. Contexto

Este estudo de caso utilizou dados provenientes dos modelos de processos e dos padrões que descrevem em detalhes os processos da organização. Estes documentos descrevem as atividades, papéis e artefatos do processo de desenvolvimento de software. Neste estudo, não foi utilizado o processo completo da empresa, pois foi selecionada especificamente uma metodologia de desenvolvimento – Orientação a Objetos – entre as aproximadamente 12 metodologias de desenvolvimento existentes na organização. Porém, para que a LPS resultante tenha um sentido completo, também foram adotadas a metodologia de gerência de projetos tradicional, a metodologia de gerência de projetos com métodos ágeis e os subprocessos de Testes e Gerência de Configuração de Software.

O participante do estudo foi o gestor do GDPS da empresa. Como responsável e especialista em definição de processos de software, ele era a pessoa mais indicada para participar deste estudo.

A partir dos processos da empresa, foi criada pela pesquisadora uma LPS inicial. Esta linha foi criada seguindo as etapas de Engenharia de Domínio do Processo de Software da sistemática da COMPOOTIM, apresentada no Capítulo 5. A LPS criada foi apresentada ao gestor do GDPS e avaliada por ele através da técnica de revisão por pares. Esta avaliação do gestor foi feita com base na sua experiência e conhecimento para verificar todos os artefatos da LPS. A cada sessão foram avaliados os diferentes artefatos que compõem a LPS, seguindo as mesmas etapas previstas pela sistemática COMPOOTIM.

Para esta etapa do estudo de caso, não foi utilizada a ferramenta de apoio da COMPOOTIM, pois os artefatos da LPS foram também modelados na

ferramenta Odyssey e optou-se por realizar a revisão em cima destes diagramas e com o material impresso, para facilitar a consulta.

6.3.1.2. Questões e Variáveis

Para alcançar os objetivos deste estudo de caso, foi elaborada uma série de questões. Para cada questão são apresentadas as variáveis avaliadas.

a) Variáveis Dependentes

As variáveis dependentes são definidas de acordo com os objetivos e as questões que se pretende responder com o estudo.

Questão 1: A LPS pôde ser criada e avaliada seguindo as etapas estabelecidas pela sistemática da COMPOOTIM?

Esta questão pretende verificar se é possível gerar os artefatos indicados para a criação da LPS, seguindo as etapas definidas na sistemática da COMPOOTIM. Os resultados obtidos com a análise desta questão permitem avaliar a aplicabilidade da sistemática da COMPOOTIM. Para analisar esta questão são utilizadas as seguintes variáveis: a) *Total de sugestões de correção na LPS*; b) *Gravidade das sugestões de correção na LPS*; c) *Tempo de Execução*.

As três primeiras variáveis foram medidas ao longo do estudo de caso através do registro em uma planilha das sugestões de revisão apresentadas pelo participante. A variável tempo de execução foi medida durante a execução de cada sessão de revisão.

Questão 2: Quais as dificuldades encontradas durante a criação/avaliação da LPS?

Esta questão tenta investigar a corretude da LPS resultante da aplicação da sistemática da COMPOOTIM. Nesta questão são avaliadas as variáveis: a) *Grau de dificuldade na criação/avaliação da LPS*; b) *Principal dificuldade na criação/avaliação da LPS*.

Estas variáveis foram medidas através da aplicação do questionário, para capturar as dificuldades que o participante pode ter sentido na revisão da LPS. De acordo com os comentários, é possível identificar oportunidades de melhorias futuras na sistemática.

Questão 3: A aplicação da sistemática nos processos da organização resultou em uma LPS adequada?

Esta questão visa avaliar a adequabilidade da LPS resultante em relação ao seu tamanho e completude. Além disso, considerando-se a experiência do Gestor do GDPS, é possível imaginar que ele consiga julgar a qualidade e viabilidade de uso prático da nova LPS. Assim, nesta questão, são avaliadas as variáveis: a) *Grau de Satisfação com o resultado obtido*; b) *Tamanho da LPS*; c) *Completude da LPS*; d) *Potencial de Uso da LPS*; e) *Adequação do tempo*.

Todas as variáveis foram apuradas através da aplicação de questionário. O objetivo da última variável é avaliar se o tempo gasto na criação/avaliação da LPS é razoável para viabilizar sua adoção prática ou se o esforço necessário atuaria de forma negativa no dia-a-dia de trabalho.

Questão 4: Todas as etapas e artefatos da LPS são realmente necessários?

Outras variáveis que necessitam ser analisadas para evitar que seja empreendido esforço com a execução de atividades que não contribuam para a representação da LPS ou que não gerem artefatos relevantes são: a) *Grau de Relevância das etapas*; b) *Grau de Relevância dos artefatos*; c) *Grau de Explicitação das Opcionalidades e Variabilidades*.

Assim, no questionário são coletadas as impressões do participante em relação a essas variáveis.

b) Variáveis Independentes

Neste trabalho, as variáveis independentes consideradas estão relacionadas ao contexto do **Gestor do GDPS**: a) *Formação acadêmica*; b) *Experiência em Engenharia de Software*; b) *Experiência em Gerência de Projetos de Software*; d) *Experiência em Definição de Processos de Software*; e) *Experiência em Reutilização de Processos de Software*; f) *Conhecimento do processo Implementar Soluções de Software utilizado*; g) *Conhecimento da metodologia de Orientação a Objetos utilizada*; h) *Conhecimento da metodologia de Gerenciamento de Projetos utilizada*; i) *Conhecimento da metodologia de Gerenciamento de Projetos com Métodos Ágeis utilizada*; e ao contexto do **processo** envolvido no estudo de caso: j) *Tamanho do processo*; k) *Complexidade do processo*; l) *Relevância do processo*.

Em relação ao Gestor do GDPS, todas as variáveis foram coletadas através do formulário de caracterização do participante com o objetivo de levantar o seu conhecimento e experiência. As outras variáveis relacionadas ao processo buscam caracterizar os processos utilizados no estudo de caso. A intenção é observar se a sistemática é aplicável a processos de qualquer tamanho e complexidade ou se possui uma atuação limitada.

6.3.1.3. Instrumentação e Preparação

Para este estudo, foram projetados cinco instrumentos principais, seguindo um modelo similar ao que foi apresentado no estudo exploratório da Seção 6.2. Estes instrumentos estão disponíveis para consulta no site²³:

- Termo de consentimento livre e esclarecido;
- Formulário de caracterização do participante;
- Material de treinamento;
- Formulário para realização do estudo;
- Questionário de avaliação do estudo.

Neste estudo, o material de treinamento inclui a apresentação dos principais conceitos da COMPOOTIM e dos artefatos gerados. O participante pôde consultar este material durante a realização do estudo.

A avaliação foi realizada através da técnica de revisão por pares. Para esta revisão, foi criado um formulário com uma tabela para registrar as observações e comentários do especialista.

6.3.1.4. Validade do Estudo

As ameaças à validade identificadas para o presente estudo são:

a) Validade interna

- Como o estudo envolve a avaliação dos artefatos criados na LPS, trata-se de um estudo longo. Para evitar que o participante fique cansado ou desanimado, o estudo será realizado em mais de uma sessão;
- Para a análise dos dados, as informações de caracterização que o participante forneceu de si mesmo serão utilizadas. Não é possível confirmar que tais informações fornecidas estejam totalmente corretas.

²³ Os instrumentos estão disponíveis para consulta em: <http://reuse.cos.ufrj.br/cdssoft/compootim-estudo-criacaolinha.html>

b) Validade externa

- Uma ameaça à validade externa é o fato de o experimento considerar apenas a LPS de uma única empresa e sob o ponto de vista de um único avaliador. Assim, não é possível generalizar os resultados para outros contextos, mas deve-se considerar que se trata de uma grande empresa brasileira com abrangência nacional.

c) Validade de construto

- Como o participante foi escolhido por conveniência, é possível que ele baseie o seu comportamento nas suposições sobre os resultados esperados por este estudo. Uma seleção aleatória dos participantes não foi possível visto que eram necessários participantes com conhecimento sobre o processo de desenvolvimento de software da empresa e com experiência na definição de processos de software;
- Outra ameaça a validade é o fato da LPS ter sido criada pela pesquisadora e ter sido validada pelo participante. Apesar de ter seguido todas as etapas da sistemática da COMPOOTIM durante esta avaliação, o participante não criou de fato a LPS desde o início

d) Validade de conclusão

- A principal ameaça à validade de conclusão neste estudo é o tamanho da amostra, com apenas uma LPS e um participante, não sendo o ideal do ponto de vista estatístico. Por isso, este estudo apresenta uma limitação quanto aos resultados, que serão considerados apenas como indícios.

6.3.2. Execução

A primeira etapa do estudo de caso foi realizada em março de 2013 com o gestor do GDPS. O perfil do participante (Tabela 6.8) indica muitos anos de experiência com Engenharia de Software, Gerenciamento de Projetos, e Definição de Processos de Software. Porém, o participante não possui experiência em Reutilização de Processos de Software. Em relação aos processos de software da organização, o participante afirmou ter muita familiaridade com todos eles (Tabela 6.9).

A execução do estudo foi realizada em quatro sessões individuais, distribuídas ao longo de um mês e com uma duração média de 01:20 hs cada. Durante as sessões, a LPS criada foi apresentada ao gestor do GDPS e

avaliada por ele através da técnica de revisão por pares. Esta avaliação resultou em 44 comentários registrados na Tabela 6.10.

Tabela 6.8 – Caracterização do Participante

Formação/Experiência		Experiência
Formação Acadêmica	Mestrado concluído	-
Experiência em Engenharia de Software	5	10 anos
Experiência em Gerência de Projetos	4	10 anos
Experiência em Definição de Processos de Software	5	10 anos
Experiência em Reutilização de Processos de Software	0	0

Legenda: 0 = nenhum; 1 = conhecimento teórico; 2 = conhecimento teórico aplicado ao contexto acadêmico; 3 = conhecimento teórico somado de experiências práticas individuais; 4 = conhecimento teórico somado de poucas experiências práticas reais; 5 = experiências práticas reais

Tabela 6.9 – Conhecimento do Processo de Software da Organização

Processo	Familiaridade
Processo Implementar Soluções de Software	Alta
Metodologia de Orientação a Objetos	Alta
Metodologia de Gerenciamento de Projetos	Alta
Metodologia de Gerenciamento de Projetos com Métodos Ágeis	Alta
Processo Implementar Soluções de Software	Alta
Metodologia de Orientação a Objetos	Alta

A Tabela 6.10 classifica a gravidade de cada observação, relaciona à fase do processo de desenvolvimento, indica em qual tipo de artefato foi feito o comentário e detalha especificamente em qual item se aplica a observação. Além disso, ao final, foi incluída uma coluna para classificar o comentário de acordo com uma adaptação da taxonomia proposta por (MELLO *et al.*, 2012).

A análise dos defeitos encontrados, resumida pelos gráficos da Figura 6.2, permite observar que a grande maioria dos defeitos tem gravidade baixa, seguida pela gravidade média. Apenas 3 defeitos tiveram gravidade alta, a saber:

- **ID1** – No caso do desenvolvimento ágil, a criação de características para as fases do desenvolvimento não foi considerada uma decisão adequada, pois isto não estava representado explicitamente no modelo do processo original. Porém, sem o uso de um agregador deste tipo, a complexidade do modelo de características aumenta, pois não fica clara a organização das características. Então, neste caso, teve que ser tomada uma decisão da

complexidade versus a fidedignidade na representação e, no modelo final, optou-se por manter a divisão por fases;

- **ID2** – Ao observar o primeiro diagrama de características, correspondente à fase de Iniciação, o participante sentiu falta de uma representação para a sequência das atividades do processo. Este item também resolveria a necessidade de um agregador no item anterior. Porém, a notação *OdysseyProcess-FEX* (em sua versão atual), adotada no Projeto CDSOft e na presente tese, não possui este tipo de representação;
- **ID7** – Por fim, o participante observou que o mesmo artefato é entrada em quase todas as tarefas. Logo, ele poderia ser colocado como entrada da atividade de Iniciação, que agrega as tarefas, ao invés de ser repetido em todas elas. Porém, novamente, esta é uma limitação da notação *OdysseyProcess-FEX* que não permite atualmente este tipo de relacionamento entre artefato e atividade.

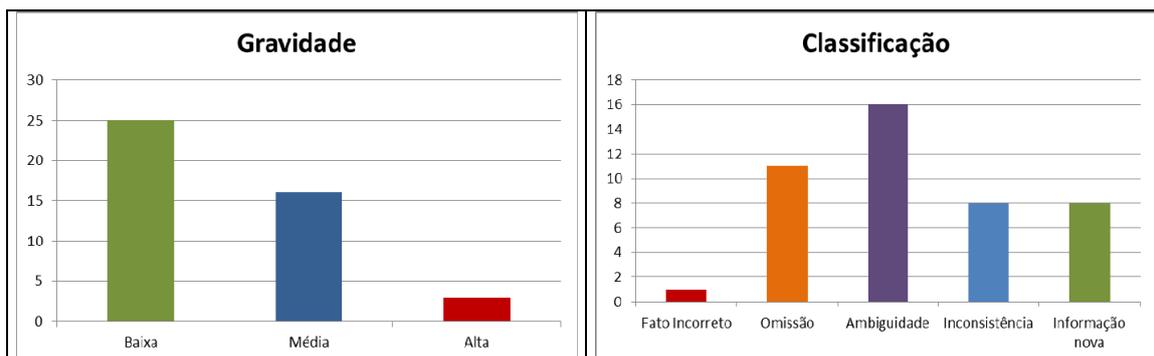


Figura 6.2 – Análises dos defeitos encontrados

De acordo com a classificação aplicada (Figura 6.2), apenas 1 defeito (ID1) foi classificado como “Fato Incorreto”, conforme explicado anteriormente. Os 11 defeitos de “Omissão” foram provocados pela ausência de algum item na LPS pela pesquisadora ou pela carência de informações sobre a obrigatoriedade dos elementos na documentação dos processos. A maioria dos defeitos foi provocada por “Ambiguidades” ou “Inconsistências” entre as diferentes documentações dos processos e precisaram ser interpretados pelo gestor do GDPS durante a execução do estudo. Os comentários classificados como “Informação Nova” dizem respeito aos itens, principalmente relacionados às regras e ao contexto, que não constavam na documentação originalmente entregue e foram acrescentados através de material suplementar pelo gestor do GDPS durante o estudo.

Tabela 6.10 – Revisão por Pares

ID	Gravidade	Fase	Tipo	Item	Comentário	Classificação
1	Alta	Iniciação	Característica	Iniciação	Não se aplica ao ágil, pois este não tem agregador. Avaliar complexidade x fidedignidade	Fato Incorreto
2	Alta	Iniciação	Característica	Geral	Sente falta da ideia de sequencia	Omissão
3	Baixa	Iniciação	Característica	Iniciar atendimento da demanda	Incluir como saída Informações do projeto de software	Omissão
4	Baixa	Iniciação	Característica	Iniciar projeto	Incluir como entrada e saída Informações do projeto de software	Omissão
5	Média	Iniciação	Característica	Levantar requisitos	Separar visão do projeto de documento visão	Ambiguidade
6	Baixa	Iniciação	Característica	Levantar requisitos	Incluir como entrada o registro da necessidade	Omissão
7	Alta	Iniciação	Característica	Geral	O mesmo artefato Registro de necessidade é entrada em quase todas as tarefas. Não poderia ser entrada da atividade Iniciação como um todo?	Omissão
8	Média	Planejamento	Característica	Planejar projeto de software	Separar visão do projeto de documento visão	Ambiguidade
9	Baixa	Planejamento	Característica	Planejar projeto de software	Incluir a lista de correções como entrada	Omissão
10	Baixa	Planejamento	Característica	Planejar projeto de software	Plano de gerenciamento de projeto é composto por cronograma e orçamento	Inconsistência
11	Baixa	Planejamento	Característica	Planejar solução de software	Incluir como entrada o documento visão	Omissão
12	Baixa	Planejamento	Característica	Planejar solução de software	Trocar o cronograma do projeto para o Plano de Gerenciamento de Projeto (PGP) como saída	Inconsistência
13	Baixa	Planejamento	Característica	Planejar sprint	Excluir o backlog da sprint	Inconsistência
14	Baixa	Monitoramento e Controle	Característica	Controlar projeto tradicional	Plano de gerenciamento de projeto é composto por cronograma e orçamento	Inconsistência
15	Baixa	Monitoramento	Característica	Acompanhar projeto ágil	Trocar para o gráfico burnup do produto	Inconsistência

ID	Gravidade	Fase	Tipo	Item	Comentário	Classificação
		e Controle				
16	Média	Execução	Característica	Detalhar iteração	Separar visão do projeto de documento visão	Ambiguidade
17	Baixa	Execução	Característica	Projetar testes	Colocar projeto lógico e físico como uma agregação de projeto de software	Inconsistência
18	Média	Execução	Característica	Especificar requisitos	Separar visão do projeto de documento visão	Ambiguidade
19	Baixa	Execução	Característica	Administrar dados	Colocar projeto lógico e físico como uma agregação de projeto de software	Inconsistência
20	Baixa	Execução	Característica	Implementar software	Colocar projeto lógico e físico como uma agregação de projeto de software	Inconsistência
21	Média	Execução	Característica	Homologar software	Separar visão do projeto de documento visão	Ambiguidade
22	Baixa	Execução	Característica	Realizar demonstração	Incluir como saída o documento de visão	Omissão
23	Baixa	Execução	Característica	Administrar dados	Alterar de mandatório para opcional	Ambiguidade
24	Baixa	Execução	Característica	Realizar inspeção	Alterar de mandatório para opcional	Ambiguidade
25	Baixa	Execução	Característica	Projeto físico	Alterar de mandatório para opcional	Ambiguidade
26	Baixa	Execução	Característica	Projeto lógico	Alterar de mandatório para opcional	Ambiguidade
27	Média	-	Regra de Composição de Característica		Incluir nas regras de tradicional e ágil, a construção da solução de software	Omissão
28	Média	-	Regra de Composição de Característica		Criadas novas regras para solicitação de viabilidade de infraestrutura	Omissão
29	Média	-	Regra de Composição de Característica		Procurar novas regras nos slides enviados	Informação nova

ID	Gravidade	Fase	Tipo	Item	Comentário	Classificação
30	Média	-	Informação de Contexto	Tipo de desenvolvimento	Nova informação de contexto	Omissão
31	Média	-	Informação de Contexto		Ver outros critérios nos slides enviados	Informação nova
32	Baixa	-	Componente	Definir critérios de aceitação	Alterar de mandatório para opcional	Ambiguidade
33	Baixa	-	Componente	Obter aceitação das entregas do projeto	Alterar de mandatório para opcional	Ambiguidade
34	Baixa	-	Componente	Todos os componentes da metodologia OO, com exceção do Implementar caso de uso	Alterar de mandatório para opcional	Ambiguidade
35	Baixa	-	Componente	Realizar reunião de planejamento da release	Alterar de mandatório para opcional	Ambiguidade
36	Baixa	-	Componente	Demonstrar produto de software	Alterar de mandatório para opcional	Ambiguidade
37	Baixa	-	Componente	Realizar retrospectiva da sprint	Alterar de mandatório para opcional	Ambiguidade
38	Baixa	-	Componente	Auditar itens relativos à fase	Alterar de mandatório para opcional	Ambiguidade
39	Média	-	Situação de Contexto	Novo desenvolvimento	Nova situação de contexto	Informação nova
40	Média	-	Situação de Contexto	Modelagem complexa	Nova situação de contexto	Informação nova
41	Média	-	Situação de Contexto	Infraestrutura não padrão	Nova situação de contexto	Informação nova
42	Média	-	Situação de Contexto	Documentação existente	Nova situação de contexto	Informação nova
43	Média	-	Regra de Contexto	Modelagem complexa	Nova regra de contexto	Informação nova
44	Média	-	Regra de Contexto	Infraestrutura não padrão	Nova regra de contexto	Informação nova

Nas próximas seções, é apresentada a LPS resultante, ou seja, já ajustada de acordo com os comentários do gestor do GDPS.

6.3.2.1. Análise de Domínio do Processo de Software

a) Linha de Processos

Assumindo a definição de uma LPS única para a organização, onde estaria representado o processo de desenvolvimento de software com toda a sua variabilidade, foram definidas as seguintes entradas disponíveis e resultados esperados:

- **Entradas Disponíveis** – as entradas disponíveis dizem respeito aos artefatos que serão produzidos nos processos de gestão da demanda e contratação que são executados antes do processo de desenvolvimento de software: Registro da Necessidade e Contratos do Projeto;
- **Resultados Esperados** – os resultados esperados representam o mínimo que a organização espera que seja produzido em qualquer projeto de desenvolvimento de software, independente da metodologia de desenvolvimento utilizada: Código-fonte, Linha Base de Produção, Versão Instalada em produção, Termo de Aceitação da Entrega e Relatório de Fechamento de Projeto.

b) Características

A principal referência para a criação do modelo de características foi o processo de desenvolvimento de software da organização. A partir desse processo, foram extraídas, no total, 86 características. As características foram organizadas por fases (iniciação, planejamento, execução, acompanhamento e encerramento). O diagrama apresentado na Figura 6.3 ilustra o modelo de características de uma das fases. As características também foram totalizadas por tipo, opcionalidade e variabilidade, conforme apresentado na Tabela 6.11.

Tabela 6.11 – Totalização das características da LPS

Fase	Total	Por tipo	Por Opcionalidade	Por Variabilidade			
Iniciação	26	Atividade	1	Mandatário	8	Invariante	26
		Tarefa	9	Opcional	18	Variante	0
		Produto	16			Ponto Var	0
Planejamento	14	Atividade	3	Mandatário	1	Invariante	11
		Tarefa	5	Opcional	13	Variante	2
		Produto	6			Ponto Var	1
Execução	33	Atividade	4	Mandatário	8	Invariante	30
		Tarefa	15	Opcional	25	Variante	2
		Produto	14			Ponto Var	1
Monitoramento	9	Atividade	1	Mandatário	3	Invariante	6

Fase	Total	Por tipo	Por Opcionalidade	Por Variabilidade			
e Controle		Tarefa	2	Opcional	6	Variante	2
		Produto	6			Ponto Var	1
Encerramento	4	Atividade	1	Mandatário	4	Invariante	4
		Tarefa	1	Opcional	0	Variante	0
		Produto	2			Ponto Var	0
Total Geral	86						

c) Regras de Composição de Características

A partir do modelo de características da LPS (apresentado na seção anterior) foram criadas 8 regras de composição de características. Algumas destas regras são apresentadas na Tabela 6.12. Por exemplo, na RCCA8, quando é necessária “Solicitar Aprovação da Visão do Projeto” deve ser necessariamente realizada a atividade “Elaborar Visão do Projeto”.

Tabela 6.12 – Regras de composição de características

# regra	Antecedente	Tipo	Operador	Consequente
RCCA 1	Planejamento ágil OR Construir solução de software ágil	Inclusiva	<i>Requires</i>	Acompanhar projeto ágil
RCCA 5	Planejamento tradicional OR Controlar projeto tradicional	Inclusiva	<i>Requires</i>	Construir solução de software tradicional
RCCA 8	Solicitar aprovação da visão do projeto	Inclusiva	<i>Requires</i>	Elaborar visão do projeto

d) Dimensões e Informações de Contexto

Foram mantidas as dimensões de contexto propostas originalmente por Araujo *et al.* (2004) (vide Capítulo 4), pois a empresa não adota nenhuma definição específica. As informações de contexto identificadas se concentraram em apenas 4 destas dimensões: cliente/usuário, equipe, produto e projeto.

As informações de contexto foram definidas com base nos critérios para classificação da complexidade do projeto (projeto – para desenvolvimentos mais complexos; ou simples – para desenvolvimentos menos complexos). Existiam também premissas para o desenvolvimento ágil que foram transformadas em informações de contexto. O modelo de contexto com as dimensões e informações de contexto é apresentado na Figura 6.4. Ao todo, foram identificadas 24 informações de contexto. Alguns exemplos destas informações, com seus respectivos valores, estão descritos na Tabela 6.13.

Tabela 6.13 –Informações de Contexto

Dimensão	Informações de Contexto	Valores
Cliente/Usuário	Dispersão Geográfica dos Envolvidos	- Somente um cliente no mesmo estado; - Mais de um cliente geograficamente distante
Equipe	Tamanho Equipe	- <=9 integrantes; - >9 integrantes
	Conhecimento da Solução	- Equipe detém conhecimento da solução; - Equipe não detém conhecimento da solução
Projeto	Conformidade com SOX	- Sim; - Não
	Instabilidade de Escopo	- Escopo estável; - Escopo instável
	Necessidade de Aquisições e/ou Contratações	- Não; - Sim
	Quantidade de HH	- Até 700 HH; - Maior que 700 HH
	Restrições de Prazo	- Sem restrições; - Com restrições;
	Sprints	- Sim; - Não
	Número de Sprints	- <2; - >=2
	Duração dos Sprints	- <1 semana; - >4 semanas; - 1<=x<=4 semanas
	Tipo de Desenvolvimento	- Manutenção corretiva; - Manutenção emergencial; - Manutenção evolutiva / adaptativa; - Novo desenvolvimento
	Utilização de Novas Tecnologias	- Com domínio da tecnologia; - Sem domínio da tecnologia
Produto Software de	Complexidade do Modelo Lógico de Dados	- Alta; - Média; -Baixa
	Disponibilidade de Documentação	- Não; - Sim
	Impacto por Indisponibilidade	- Baixa robustez; - Alta/média robustez
	Infraestrutura Padrão	- Não; - Sim
	Necessidade de Integração	- Sem necessidade de integração; - Com necessidade de integração

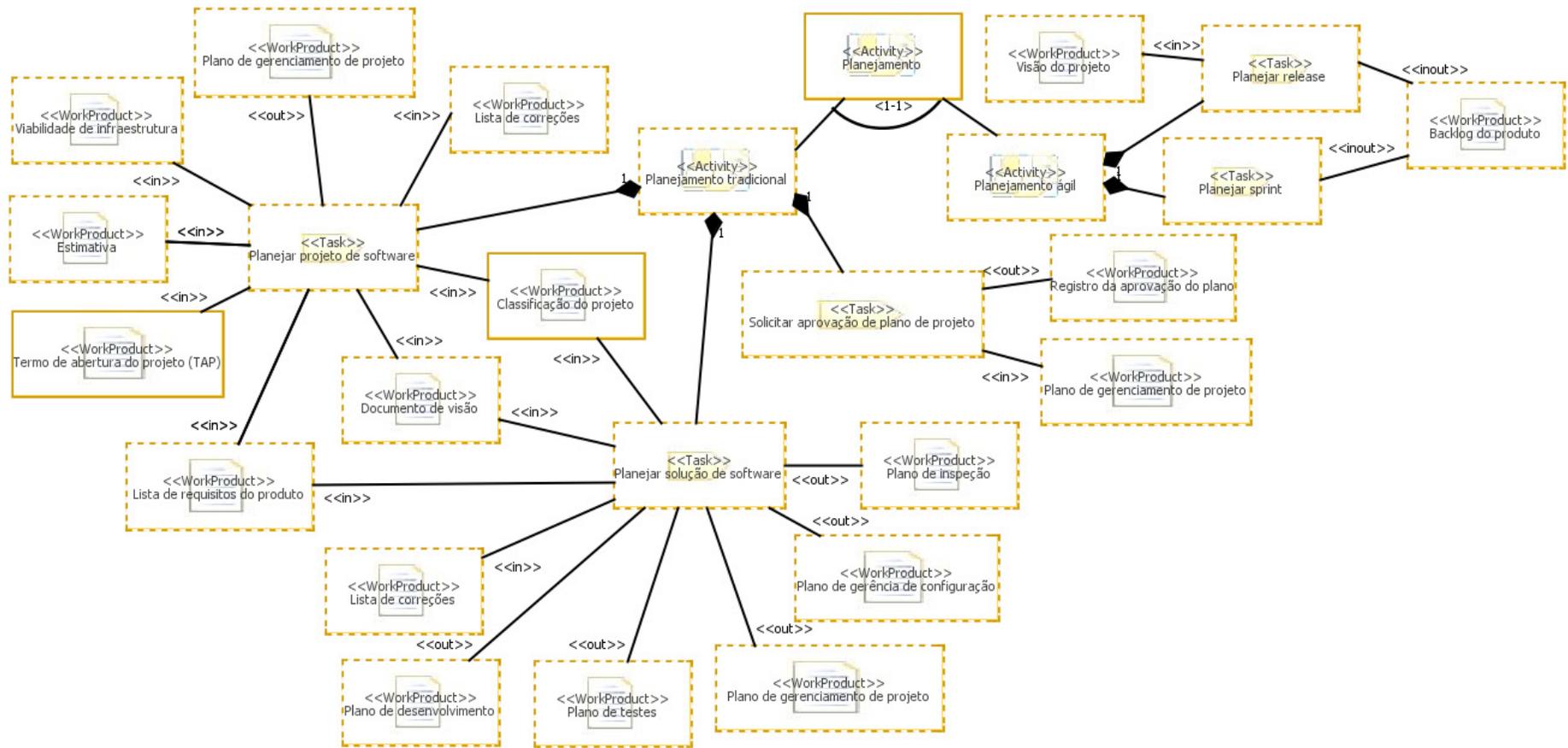


Figura 6.3 – Fase Planejamento - Modelo de características no Odyssey

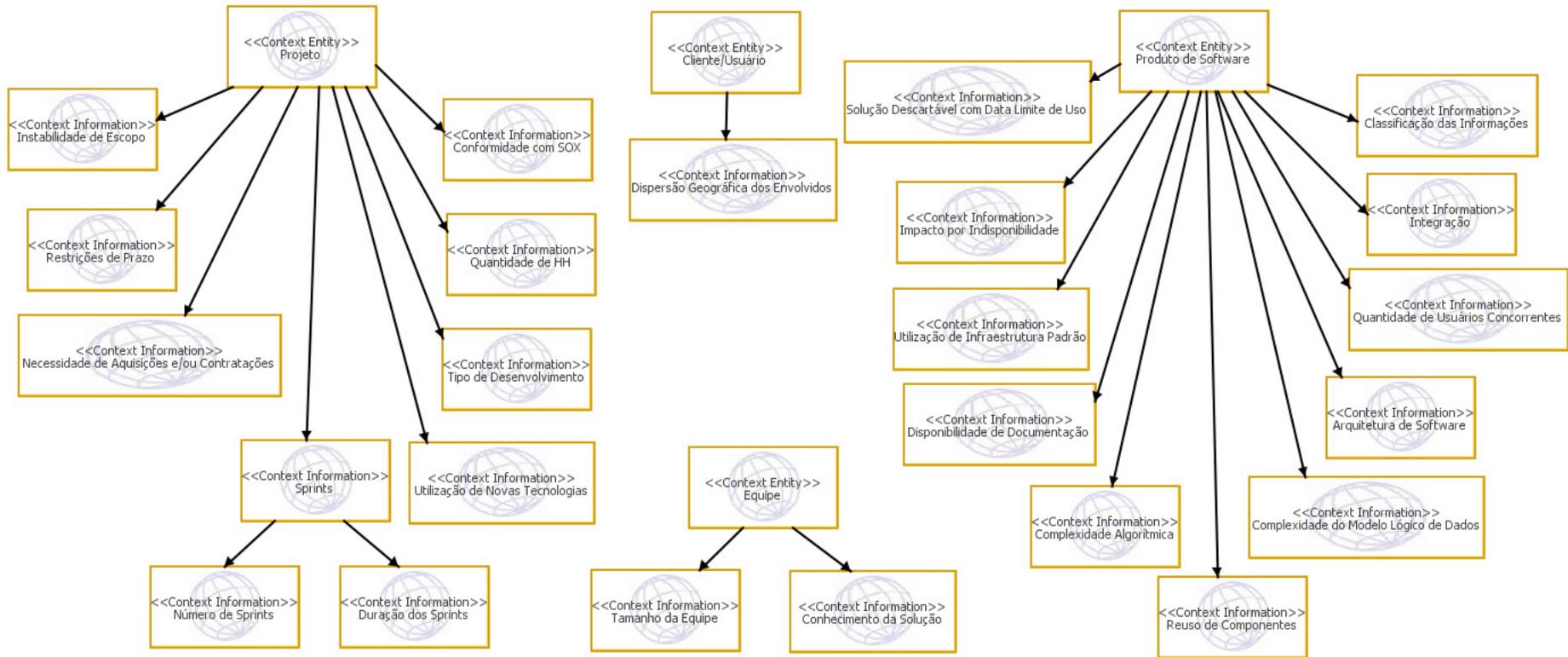


Figura 6.4 – Dimensões e Informações de Contexto modeladas no Odyssey

6.3.2.2. Projeto de Domínio do Processo de Software

a) Componentes de Processo

A partir dos processos definidos pela organização, foram extraídos 86 componentes de processo que detalham as 32 características do tipo “Tarefa” criadas na fase anterior da LPS. Os componentes de processo também foram totalizados, conforme apresentado na Tabela 6.14.

Tabela 6.14 – Totalização dos componentes da LPS

Processo			Total	Por Opcionalidade	
Metodologia Gerenciamento de Projetos		de	29	Mandatório	12
				Opcional	17
Metodologia Gerenciamento de Projetos com Métodos Ágeis		de	8	Mandatório	4
				Opcional	4
Metodologia Orientação a Objetos			21	Mandatório	1
				Opcional	20
Realizar Teste de Software			5	Mandatório	4
				Opcional	1
Gerência de Configuração de Software			12	Mandatório	8
				Opcional	4
Gerais			11	Mandatório	3
				Opcional	8
Total Geral			86	Mandatório	32
				Opcional	54

b) Regras de Composição de Componentes

A partir dos componentes de processo, foram identificadas 9 regras de composição de componentes (Tabela 6.15). Todas as regras são do tipo Inclusiva, pois nenhuma regra Exclusiva foi levantada.

Tabela 6.15 – Regras de Composição de Componentes

# regra	Antecedente	Tipo	Operador	Consequente
RCCO1	Monitorar e controlar riscos do projeto	Inclusiva	<i>Requires</i>	Elaborar plano de riscos
RCCO2	Monitorar e controlar aquisições do projeto	Inclusiva	<i>Requires</i>	Elaborar plano de aquisições
RCCO3	Documentar lições aprendidas	Inclusiva	<i>Requires</i>	Registrar lições aprendidas
RCCO4	Detalhar casos de uso	Inclusiva	<i>Requires</i>	Identificar atores e casos de uso
RCCO5	Refinar diagrama de interação	Inclusiva	<i>Requires</i>	Elaborar diagrama de interação
RCCO6	Executar testes unitários	Inclusiva	<i>Requires</i>	Implementar classes de teste unitário
RCCO7	Administrar dados	Inclusiva	<i>Requires</i>	Elaborar modelo lógico de dados
RCCO8	Colocar em homologação	Inclusiva	<i>Requires</i>	Marcar linha-base de homologação

# regra	Antecedente	Tipo	Operador	Consequente
RCCO9	Marcar linha-base de homologação	Inclusiva	<i>Requires</i>	Solicitar passagem para homologação

c) Situações de Contexto

As situações de contexto representam circunstâncias que podem acontecer baseadas na combinação de valores de determinadas informações de contexto. As situações de contexto criadas são apresentadas na Tabela 6.16.

Tabela 6.16 – Situações de Contexto

Situação de Contexto	Expressão
Projeto Ágil	Sprints=Sim AND (Número de Sprints= ≥ 2) AND (Tamanho Equipe= ≤ 9 integrantes) AND (Duração dos Sprints= $1 \leq x \leq 4$)
Projeto com Escopo Instável	Instabilidade de Escopo=Escopo instável
Projeto com Restrições de Prazo	Restrições de Prazo=Com restrições
Projeto SOX	Conformidade com SOX=Sim
Projeto com Dispersão Geográfica dos Envolvidos	Dispersão Geográfica dos Envolvidos= Mais de um cliente geograficamente distantes
Projeto Com Aquisições	Necessidade de Aquisições e/ou Contratações=Sim
Simple Sem Aquisições	Necessidade de Aquisições e/ou Contratações=Não
Projeto com Esforço Grande	Quantidade de HH > 700HH
Simple com Esforço Pequeno	Quantidade de HH ≤ 700 HH
Projeto sem Domínio da Tecnologia	Utilização de Novas Tecnologias=Sem domínio da tecnologia
Simple com Domínio da Tecnologia	Utilização de Novas Tecnologias=Com domínio da tecnologia
Projeto com Integração	Integração=Com necessidade de integração
Simple sem Integração	Integração=Sem necessidade de integração
Projeto com Impacto por Indisponibilidade	Impacto por Indisponibilidade=Alta/média robustez
Simple sem Impacto por Indisponibilidade	Impacto por Indisponibilidade=Baixa robustez
Novo desenvolvimento	Tipo de Desenvolvimento=Novo desenvolvimento
Modelagem complexa	Complexidade do Modelo Lógico de Dados=Alta
Infraestrutura não padrão	Utilização de Infraestrutura Padrão=Não
Documentação existente	Disponibilidade de Documentação=Sim

d) Regras de Contexto

As regras de contexto foram geradas com base nas situações de contexto e no modelo de características da LPS, apresentados anteriormente. No total, foram elencadas 5 regras de contexto, listadas na Tabela 6.17. Vale ressaltar que não foram definidas regras de contexto estabelecendo ações para todas as situações de contexto identificadas, o que significa que na prática algumas situações de contexto não serão aplicadas neste estudo de caso. Um maior uso das situações e regras de contexto na LPS iria aumentar a riqueza das sugestões que a COMPOOTIM será capaz de fazer na próxima etapa do estudo, quando ocorre a composição dos processos de software.

Tabela 6.17 – Regras de Contexto

# regra	Situação de Contexto	Operador	Característica
RCTX1	Projeto Ágil	<i>Implies</i>	Planejamento ágil AND Acompanhar projeto ágil AND Construir solução de software ágil AND Elaborar visão do projeto
RCTX2	Projeto com Escopo Instável AND Projeto com Restrições de Prazo AND Projeto SOX AND Projeto com Dispersão Geográfica dos Envolvidos AND Projeto com Esforço Grande AND Projeto com Impacto por Indisponibilidade	<i>Implies</i>	Planejamento tradicional AND Controlar projeto tradicional AND Construir solução de software tradicional
RCTX3	Novo desenvolvimento OR Infraestrutura não padrão OR Projeto com Aquisições	<i>Implies</i>	Avaliar viabilidade de infraestrutura
RCTX4	Projeto SOX	<i>Implies</i>	Realizar inspeção
RCTX5	Modelagem complexa	<i>Implies</i>	Administrar dados

6.3.3. Análise dos Resultados

Esta seção resume os principais resultados obtidos com a execução da primeira etapa deste estudo de caso. Os resultados foram estruturados de acordo com as variáveis avaliadas. Algumas variáveis foram coletadas através do questionário preenchido pelo participante após a avaliação da LPS.

6.3.3.1. Processos da Organização

Em relação às variáveis independentes sobre os processos originais da organização, o gestor do GDPS afirmou que os processos são grandes, com

uma complexidade média e com alta relevância para a organização (Tabela 6.18). Foi explicado que alguns membros da empresa consideram os processos complexos, mas que diversas melhorias já foram feitas e que, portanto, do ponto de vista do participante, essa complexidade já foi reduzida.

Tabela 6.18 – Processos da Organização

Item	Característica
Tamanho dos Processos	Grande
Complexidade dos Processos	Média
Relevância dos Processos	Alta

6.3.3.2. Linha de Processos Resultante

Sobre a criação e avaliação da LPS, o participante afirmou que conseguiu executar todas as tarefas e ficou satisfeito com o resultado final obtido (Tabela 6.19). Em relação à LPS resultante, o participante considerou que ela tem um tamanho grande (Tabela 6.19), devido à quantidade de artefatos que a compõem. Mas observou que *“nunca vi outra linha de processos para poder comparar o tamanho e ter uma ideia se está razoável”*. Ele afirmou ainda que a linha resultante tem uma completude alta e que o tempo para criação/avaliação da LPS pareceu razoável (Tabela 6.19).

Tabela 6.19 – Linha de Processos Resultante

Item	Resposta
Execução	Sim
Satisfação	Satisfeito
Tamanho da LPS	Grande
Completude da LPS	Alta
Tempo	Parece razoável

6.3.3.3. Dificuldades encontradas durante a criação/avaliação da LPS

Apesar de a LPS inicial ter sido criada pela pesquisadora, durante a revisão foram seguidas todas as etapas e revistos todos os artefatos da LPS. Assim, a avaliação da LPS permitiu perceber também quais seriam as dificuldades de criação da LPS. O participante avaliou como grau de dificuldade média a criação/revisão da LPS e afirmou que *“foram necessárias algumas interpretações”* (Tabela 6.20). Estas interpretações se referem ao exercício de

encaixar o processo atualmente existente na organização na nova forma de representação proposta pelo Projeto CDSOft, visto que a modelagem atual do processo é feita usando a notação EPC (*Event-Driven Process Chain*) (SCHEER, 2000) e na criação da LPS foi adotada a notação *OdysseyProcess-FEX* (TEIXEIRA, 2011) baseada em modelagem de características.

Perguntado sobre a maior dificuldade durante a avaliação da LPS, o participante afirmou que foi “*realizar o trabalho sobre regras de composição*” (Tabela 6.20). De fato, ao longo da revisão da LPS, o participante fez algumas observações sobre as dificuldades de entendimento da necessidade das regras de composição, pois algumas delas parecem óbvias e poderiam ser substituídas por mecanismos de sequenciamento dos componentes de processo. A dificuldade com as regras de composição também pode ser explicada pelo fato de que muitas vezes as regras de contexto e de composição se confundiam.

Tabela 6.20 – Dificuldades encontradas

Item	Resposta
Grau de Dificuldade	Média
Maior Dificuldade	Regras de Composição

6.3.3.4. Necessidade das Etapas e Artefatos

Visando avaliar também as etapas e artefatos propostos pela sistemática da COMPOOTIM, o participante foi questionado sobre a real necessidade da existência de todos os elementos. Ele considerou que todas as etapas são realmente necessárias, mas que apenas alguns artefatos são necessários (Tabela 6.21), pois entende “*não haver necessidade das regras de composição (a maioria das regras pareciam óbvias)*”, conforme discutido na seção anterior.

Tabela 6.21 – Necessidade das etapas e artefatos

Item	Resposta
Necessidade das Etapas	Todas as etapas são necessárias
Necessidade dos Artefatos	Alguns artefatos são necessários
Opcionalidades e Variabilidades	Parcialmente explícitas

Também foi analisado se a sistemática contribuiu para a explicitação das opcionalidades e variabilidades (Tabela 6.21). O participante indicou que elas ficaram parcialmente explícitas e acrescentou: “*Não por limitação da*

representação, mas se fosse mapeado o processo completo da organização, não seria possível garantir a corretude do modelo sem o apoio de uma ferramenta, devido à quantidade de informações representadas nos modelos". Este comentário pode ser entendido considerando que durante o estudo de caso foi utilizada somente a metodologia de desenvolvimento Orientação a Objetos, dentre as diversas metodologias de desenvolvimento atualmente adotadas pela organização. Com a modelagem das demais metodologias, inevitavelmente a LPS gerada seria maior. Neste ponto, observa-se a preocupação do participante com o entendimento da representação completa do processo de software da empresa (Tabela 6.18).

6.3.3.5. Vantagens, Desafios e Melhorias

Considerando-se a experiência do participante como gestor do GDPS, é possível imaginar que ele consiga julgar a viabilidade de uso prático da LPS criada na organização. O participante afirmou que a LPS poderia ser adotada na organização com adaptações (Tabela 6.22). As adaptações referidas vêm da ideia, apresentada pelo participante, de que em uma sistemática seria possível escolher os artefatos que serão construídos visando atingir o resultado final (a construção da LPS), mas sem ter que elaborar todos os modelos ou regras caso eles não sejam necessários em uma determinada organização. Esta possibilidade precisaria ser investigada em outras avaliações futuras da sistemática da COMPOOTIM para validar se ela é capaz de atingir os resultados esperados mesmo sem a adoção de um ou mais artefatos.

Ainda ao se referir à viabilidade de uso prático da LPS, foi destacado espontaneamente pelo participante: *"Sim e solicitei o modelo de características com as totalizações para uso em trabalho na empresa"*. Este comentário ressalta uma contribuição prática da COMPOOTIM para as discussões internas dos processos na empresa.

A principal vantagem identificada pelo participante foi o fato da sistemática ajudar a avaliar a completude e identificar inconsistências no processo (Tabela 6.22). Essa possibilidade de analisar o processo também foi elencada como um aspecto positivo da COMPOOTIM. Este benefício pode ser considerado um efeito colateral inesperado, pois não tinha sido previsto durante a proposição da COMPOOTIM. Desta forma, como trabalhos futuros, podem ser desenvolvidos outros mecanismos que facilitem essas verificações

de consistências.

Por outro lado, o principal desafio mencionado foi a dificuldade em manter a LPS atualizada, visto que os processos da empresa sofrem alterações frequentes (Tabela 6.22). Como existem diversos artefatos com rastreabilidade entre si, a preocupação do gestor do GDPS é que as manutenções sejam custosas caso as alterações tenham que ser propagadas manualmente entre os diferentes artefatos. Porém, vale ressaltar que esta etapa do estudo foi conduzida sem o uso do apoio computacional provido pela COMPOOTIM. Realmente, caso tivessem que ser realizadas de forma totalmente manual, estas atualizações seriam muito trabalhosas, mas a COMPOOTIM pode facilitar estas manutenções.

Tabela 6.22 – Vantagens, Desafios e Melhorias

Item	Resposta
LPS criada poderia ser utilizada na organização	Sim, poderia ser utilizada com adaptações
Vantagens	Utilizar uma nova abordagem para poder validar a completude do processo e inconsistências
Desafios	Manter atualizada a linha de processo em função das constantes alterações que fazemos no processo
Aspectos Positivos	A possibilidade de analisar o processo e otimizá-lo
Aspectos Negativos	Não é simples, trabalhosa e, algumas vezes, difícil de ser executada sem o auxílio do pesquisador
Sugestões de Melhoria	Precisa ser avaliado se todos os modelos utilizados para linhas de produto estão/são adequados para linha de processos
Comentários Adicionais	Precisamos investir mais no tema para alcançarmos a melhor abordagem dentro do conceito que está sendo proposto de linhas de processo

Como aspectos negativos foram listados a falta de simplicidade da sistemática que pode se tornar trabalhosa e difícil de ser executada sem auxílio (do pesquisador e de uma ferramenta). O participante explicou essas afirmações da seguinte forma: *“Trabalhosa por ser uma abordagem nova, que adota uma notação diferente”*. Apesar de o participante reconhecer a necessidade de todas as etapas (vide seção anterior), elas ainda são consideradas trabalhosas. A necessidade do apoio do pesquisador vem da *“dificuldade em adquirir conhecimento novo para fazer sozinho quando já se está acostumado a trabalhar de outra forma”*. Este comentário remete à

dificuldade de mudança de paradigma, visto que nas organizações tipicamente são adotadas outras notações (BPMN – *Business Process Management Notation* (OMG, 2009), EPC (SCHEER, 2000) e SPEM (OMG, 2008)) para representação dos processos. Além disso, as estruturas de reutilização de processos, como os componentes de processos, são uma novidade para o participante, que ainda não tinha experiência na área. A demanda pelo apoio computacional é explicada pelo fato da ferramenta COMPOOTIM não ter sido utilizada nesta etapa do estudo com o propósito de evitar desviar o foco da avaliação do conteúdo da LPS.

Como sugestão de melhoria, o participante citou a necessidade de se avaliar se todos os modelos utilizados são realmente adequados para LPS (Tabela 6.22), novamente remetendo às dificuldades com as regras de composição e que talvez, em algumas situações, pudessem ser substituídas por uma representação típica de modelagem de processos para o sequenciamento dos componentes de processos. Por fim, nos comentários adicionais, o participante reforçou o interesse da organização em dar prosseguimento aos trabalhos de pesquisa conjuntos no tema (Tabela 6.22).

6.3.4. Conclusões e Limitações

Após a realização desta primeira etapa do estudo, uma LPS completa e com todos os artefatos previstos na sistemática ficou pronta para ser utilizada na etapa seguinte do estudo de caso, onde o propósito é avaliar a composição dos processos pela COMPOOTIM. Além deste resultado concreto, esta primeira etapa do estudo contribuiu para a avaliação das etapas e artefatos propostos pela sistemática para a criação da LPS.

As principais conclusões obtidas durante esta avaliação dizem respeito à viabilidade de uso prático da LPS, que contribuiu para as discussões internas dos processos na empresa. Observou-se que a sistemática contribuiu para avaliar a completude e identificar inconsistências no processo.

Em relação à notação de representação dos processos, a *OdysseyProcess-FEX* representa uma mudança de paradigma para as organizações acostumadas com notações típicas de modelagem de processos. Em particular, a ausência do sequenciamento no processo é uma limitação atual da *OdysseyProcess-FEX*. O uso de regras de composição também

precisa ser revisto, pois foi uma dificuldade para o gestor do GDPS.

Sobre as dificuldades com as regras de composição, além das questões técnicas já discutidas durante a análise dos resultados, vale ressaltar também que os processos da organização não oferecem um alto nível de variabilidade. De fato, no nível dos componentes de processo não foi identificada nenhuma variabilidade. Isto ocorre porque atualmente a definição de processos da empresa está baseada na adoção de um processo padrão genérico. Como não é desejado pela empresa sobrecarregar os gerentes de projeto com muitas decisões durante a adaptação do processo específico para o projeto a partir do processo padrão da organização, o processo tem um baixo nível de flexibilidade, o que limita a atuação das regras de composição.

Como ainda não existe modularidade nos processos através de estruturas de reutilização, o participante não foi capaz de identificar muitas variabilidades no processo. A reutilização de processos é um tema ainda novo na empresa e este conhecimento não está disseminado. Este estudo de caso é um passo neste sentido. A motivação da empresa em disponibilizar um profissional de nível gerencial para participar do estudo de caso e o interesse da empresa no tema demonstra que se trata de um problema real da indústria de software.

Ainda que esta etapa tenha obtido resultados positivos, vale ressaltar que se trata apenas de uma evidência inicial, devido ao número restrito de participantes e ao uso somente de parte dos processos da organização. Porém, o interesse da empresa em manter um esforço de pesquisa conjunto, representa uma oportunidade concreta para a ampliação futura da LPS criada.

6.4. Estudo de Caso – Etapa 2 – Avaliação da Composição dos Processos

Esta seção descreve a etapa de avaliação da composição de processos específicos para os projetos, a partir da LPS criada na etapa anterior.

6.4.1. Planejamento

Esta seção descreve o planejamento²⁴ da Etapa 2 , cujo propósito principal é

²⁴ O plano completo da etapa 2 do estudo de caso pode ser obtido em: <http://reuse.cos.ufrj.br/cdsoft/comootim-estudo-usolinha.html>

avaliar se a COMPOOTIM é capaz de compor um processo de acordo com o contexto do projeto.

Analisar a sistemática COMPOOTIM com o ferramental de apoio construído

Com o propósito de caracterizar

Com respeito à viabilidade de composição do processo de software de acordo com o contexto do projeto

Do ponto de vista do gestor do GDPS

No contexto de um ambiente real da indústria

6.4.1.1. Contexto

Esta fase do estudo de caso utilizou a LPS criada e avaliada na fase anterior e dois projetos reais de desenvolvimento de software da organização. Os projetos foram selecionados aleatoriamente pelo próprio gestor do GDPS e caracterizados por ele de acordo com as informações de contexto (Seção 6.3.2.1) adotadas pela organização.

A partir daí, foi realizada pela COMPOOTIM a composição dos processos para cada um dos projetos indicados (de acordo com as técnicas de composição e otimização implementadas na COMPOOTIM e descritas no capítulo anterior). Os processos compostos foram apresentados ao gestor do GDPS e avaliados por ele.

6.4.1.2. Questões e Variáveis

Para alcançar os objetivos desta etapa do estudo de caso, elaborou-se uma série de questões. Para cada questão são apresentadas as variáveis avaliadas.

a) Variáveis Dependentes

As variáveis dependentes são definidas de acordo com os objetivos e as questões que se pretende responder com o estudo.

Questão 1: Um processo pode ser composto para o projeto seguindo a sistemática e usando o ferramental de apoio da COMPOOTIM?

Esta questão pretende verificar a viabilidade da COMPOOTIM, ou seja, se ela é capaz de produzir um processo composto para o projeto. Para analisar esta questão são utilizadas as seguintes variáveis: a) *Adequabilidade do processo*

composto; b) Grau de Satisfação com o resultado obtido; c) Tamanho relativo do processo composto; d) Completude do processo composto.

A primeira variável foi medida através da observação do material produzido ao final desta etapa do estudo de caso. As demais variáveis foram medidas através da aplicação de questionário.

Questão 2: Quais as dificuldades encontradas com os processos sugeridos pela composição do processo?

Esta questão tenta investigar as dificuldades percebidas durante a aplicação da COMPOOTIM. Além disso, considerando-se a experiência do Gestor do GDPS, é possível imaginar que ele consiga julgar a qualidade e viabilidade de uso prático dos processos sugeridos. Assim, nesta questão serão avaliadas as variáveis: a) *Grau de dificuldade na composição do processo; b) Principal dificuldade na composição do processo; c) Total de sugestões de correção do processo; d) Gravidade das sugestões de correção no processo; e) Potencial de uso do processo composto.*

As duas primeiras variáveis foram medidas através da aplicação do questionário, para capturar as dificuldades que o participante percebeu durante a análise dos resultados da composição do processo. As variáveis relacionadas às sugestões de correção no processo dizem respeito às possíveis observações do participante em relação às necessidades de ajuste no processo composto. Estas observações foram capturadas ao longo do estudo de caso através do registro em uma planilha.

b) Variáveis Independentes

Nesta etapa, as variáveis independentes consideradas estão relacionadas ao contexto dos **Projetos** utilizados no estudo de caso: a) *Tamanho do projeto; b) Criticidade do projeto.* A intenção é observar se a COMPOOTIM é aplicável a projetos de qualquer tamanho e criticidade ou se possui uma atuação limitada.

6.4.1.3. Instrumentação e Preparação

Para esta parte do estudo, foram projetados dois instrumentos principais, disponíveis para consulta no site²⁵:

- Formulário de caracterização de projeto;

²⁵ Os instrumentos estão disponíveis para consulta em: <http://reuse.cos.ufrj.br/cdsoft/compootim-estudo-usolinha.html>

- Questionário de avaliação do estudo.

O primeiro instrumento, formulário de caracterização de projeto, permite ao pesquisador capturar as informações de contexto de cada projeto, o que auxilia na análise posterior dos dados obtidos por meio do estudo.

Após a composição dos processos do projeto, os resultados obtidos foram avaliados com o participante através de um questionário de avaliação do estudo, que permite registrar as observações e comentários do especialista.

6.4.1.4. Validade do Estudo

Além das ameaças já identificadas na etapa anterior do estudo, as seguintes ameaças à validade foram identificadas para esta etapa:

a) Validade interna

- Para a análise dos dados, as informações de caracterização do projeto que o participante forneceu serão utilizadas. Não é possível confirmar que tais informações fornecidas estejam corretas. Porém, mesmo que as informações de caracterização do projeto não representem totalmente a realidade do projeto em questão, elas representam um tipo possível de projeto realizado pela empresa. Além disso, como a COMPOOTIM deve ser capaz de compor processos para diferentes projetos, até mesmo o uso de projetos simulados seria possível, ainda que com limitações.

b) Validade de construto

- De acordo com a sistemática da COMPOOTIM, esta etapa de composição de processos deveria ser realizada pelo Gerente de Projetos. Para que isso fosse possível, seria necessário que os gerentes de projetos da organização estivessem familiarizados com os conceitos e o conteúdo da LPS criada. Dado o tempo disponível, neste estudo não foi possível envolver os Gerentes de Projetos da organização. Então, optou-se por utilizar como participante do estudo, o próprio gestor do GDPS que possui um conhecimento mais abrangente sobre os processos da organização e já tinha participado da etapa anterior deste estudo de caso. Como responsável pela definição dos processos da organização e especialista em definição de processos de software, ele conhece diversos projetos da organização e está apto a avaliar os processos compostos para os projetos.

6.4.2. Execução

Seguindo a sistemática da COMPOOTIM, o primeiro passo para a realização desta etapa do estudo de caso é a caracterização dos projetos. Este passo foi realizado em abril de 2013 em uma única sessão com o participante.

O Projeto A segue um modelo de desenvolvimento de software orientado ao planejamento e tem duração entre 06 meses e 1 ano. As informações de contexto do Projeto A foram caracterizadas pelo participante conforme apresentado na Tabela 6.23.

Tabela 6.23 – Caracterização do Projeto A

Dimensão	Informações de Contexto	Valores
Cliente/Usuário	Dispersão Geográfica dos Envolvidos	<input type="checkbox"/> Somente um cliente no mesmo estado
		<input checked="" type="checkbox"/> Mais de um cliente geograficamente distantes
Equipe	Tamanho Equipe	<input type="checkbox"/> ≤9 integrantes
		<input checked="" type="checkbox"/> >9 integrantes
Equipe	Conhecimento da Solução	<input checked="" type="checkbox"/> Equipe detém conhecimento da solução
		<input type="checkbox"/> Equipe não detém conhecimento da solução
Projeto	Conformidade com SOX	<input checked="" type="checkbox"/> Sim
		<input type="checkbox"/> Não
	Instabilidade de Escopo	<input type="checkbox"/> Escopo estável
		<input checked="" type="checkbox"/> Escopo instável
	Necessidade de Aquisições e/ou Contratações	<input checked="" type="checkbox"/> Não
		<input type="checkbox"/> Sim
	Quantidade de HH	<input type="checkbox"/> Até 700 HH
		<input checked="" type="checkbox"/> Maior que 700 HH
	Restrições de Prazo	<input type="checkbox"/> Sem restrições
		<input checked="" type="checkbox"/> Com restrições
	Sprints	<input type="checkbox"/> Sim
		<input checked="" type="checkbox"/> Não
	Número de Sprints	<input type="checkbox"/> <2
		<input type="checkbox"/> ≥2
Duração dos Sprints	<input type="checkbox"/> <1 semana	
	<input type="checkbox"/> >4 semanas	
	<input type="checkbox"/> 1 ≤ x ≤ 4 semanas	
Tipo de Desenvolvimento	<input type="checkbox"/> Manutenção corretiva	
	<input type="checkbox"/> Manutenção emergencial	
	<input checked="" type="checkbox"/> Manutenção evolutiva / adaptativa	
	<input type="checkbox"/> Novo desenvolvimento	
Utilização de Novas Tecnologias	<input checked="" type="checkbox"/> Com domínio da tecnologia	
	<input type="checkbox"/> Sem domínio da tecnologia	

Dimensão	Informações de Contexto	Valores
Produto de Software	Arquitetura de Software	<input type="checkbox"/> Integração
		<input checked="" type="checkbox"/> Padrão
	Classificação das Informações	<input type="checkbox"/> Informações classificadas como confidencial
		<input checked="" type="checkbox"/> Informações classificadas como corporativa
		<input type="checkbox"/> Informações classificadas como pública
		<input type="checkbox"/> Informações classificadas como reservada
	Complexidade Algorítmica	<input type="checkbox"/> Alta
		<input type="checkbox"/> Média
		<input checked="" type="checkbox"/> Baixa
	Complexidade do Modelo Lógico de Dados	<input checked="" type="checkbox"/> Alta
		<input type="checkbox"/> Média
		<input type="checkbox"/> Baixa
	Disponibilidade de Documentação	<input checked="" type="checkbox"/> Não
		<input type="checkbox"/> Sim
	Impacto por Indisponibilidade	<input type="checkbox"/> Baixa robustez
		<input checked="" type="checkbox"/> Alta/média robustez
	Integração	<input checked="" type="checkbox"/> Sem necessidade de integração
		<input type="checkbox"/> Com necessidade de integração
	Quantidade de Usuários Concorrentes	<input type="checkbox"/> ≤300
		<input checked="" type="checkbox"/> >300
Reuso de Componentes	<input type="checkbox"/> Alto grau de reuso	
	<input checked="" type="checkbox"/> Médio grau de reuso	
	<input type="checkbox"/> Baixo grau de reuso	
	<input type="checkbox"/> Nenhum grau de reuso	
Solução Descartável com Data Limite de Uso	<input checked="" type="checkbox"/> Não	
	<input type="checkbox"/> Sim	
Utilização de Infraestrutura Padrão	<input type="checkbox"/> Não	
	<input checked="" type="checkbox"/> Sim	

O Projeto B segue um modelo de desenvolvimento de software ágil e também tem duração entre 06 meses e 1 ano. As informações de contexto do Projeto B foram caracterizadas pelo participante conforme apresentado na Tabela 6.24.

Tabela 6.24 – Caracterização do Projeto B

Dimensão	Informações de Contexto	Valores
Cliente/Usuário	Dispersão Geográfica dos Envolvidos	<input type="checkbox"/> Somente um cliente no mesmo estado
		<input checked="" type="checkbox"/> Mais de um cliente geograficamente distantes

Dimensão	Informações de Contexto	Valores
Equipe	Tamanho Equipe	(X) <=9 integrantes
		() >9 integrantes
	Conhecimento da Solução	() Equipe detém conhecimento da solução
		(X) Equipe não detém conhecimento da solução
Projeto	Conformidade com SOX	() Sim
		(X) Não
	Instabilidade de Escopo	() Escopo estável
		(X) Escopo instável
	Necessidade de Aquisições e/ou Contratações	(X) Não
		() Sim
	Quantidade de HH	(X) Até 700 HH
		() Maior que 700 HH
	Restrições de Prazo	() Sem restrições
		(X) Com restrições
	Sprints	(X) Sim
		() Não
	Número de Sprints	() <2
		(X) >=2
Duração dos Sprints	() <1 semana	
	() >4 semanas	
	(X) 1<=x<=4 semanas	
Tipo de Desenvolvimento	() Manutenção corretiva	
	() Manutenção emergencial	
	() Manutenção evolutiva / adaptativa	
	(X) Novo desenvolvimento	
Utilização de Novas Tecnologias	(X) Com domínio da tecnologia	
	() Sem domínio da tecnologia	
Produto de Software	Arquitetura de Software	(X) Integração
		() Padrão
	Classificação das Informações	() Informações classificadas como confidencial
		(X) Informações classificadas como corporativa
		() Informações classificadas como pública
		() Informações classificadas como reservada
	Complexidade Algorítmica	() Alta
		(X) Média
		() Baixa
	Complexidade do Modelo Lógico de Dados	() Alta
		(X) Média
() Baixa		
Disponibilidade de	(X) Não	

Dimensão	Informações de Contexto	Valores
	Documentação	() Sim
	Impacto por Indisponibilidade	() Baixa robustez
		(X) Alta/média robustez
	Integração	() Sem necessidade de integração
		(X) Com necessidade de integração
	Quantidade de Usuários Concorrentes	(X) <=300
		() >300
	Reuso de Componentes	() Alto grau de reuso
		(X) Médio grau de reuso
		() Baixo grau de reuso
		() Nenhum grau de reuso
	Solução Descartável com Data Limite de Uso	(X) Não
		() Sim
	Utilização de Infraestrutura Padrão	() Não
(X) Sim		

Em seguida, os dados da caracterização dos projetos foram cadastrados na ferramenta COMPOOTIM, onde já constavam os dados da LPS criada. A partir destes dados, a ferramenta determinou as situações e regras de contexto (Seção 6.3.2.2) que se aplicavam a cada um dos projetos, conforme ilustrado com o Projeto B na Figura 6.5. O Projeto A foi tipicamente caracterizado como um projeto tradicional da organização, enquanto o Projeto B está orientado ao desenvolvimento ágil. Esta diversidade dos modelos de desenvolvimento traz maior riqueza a este estudo.

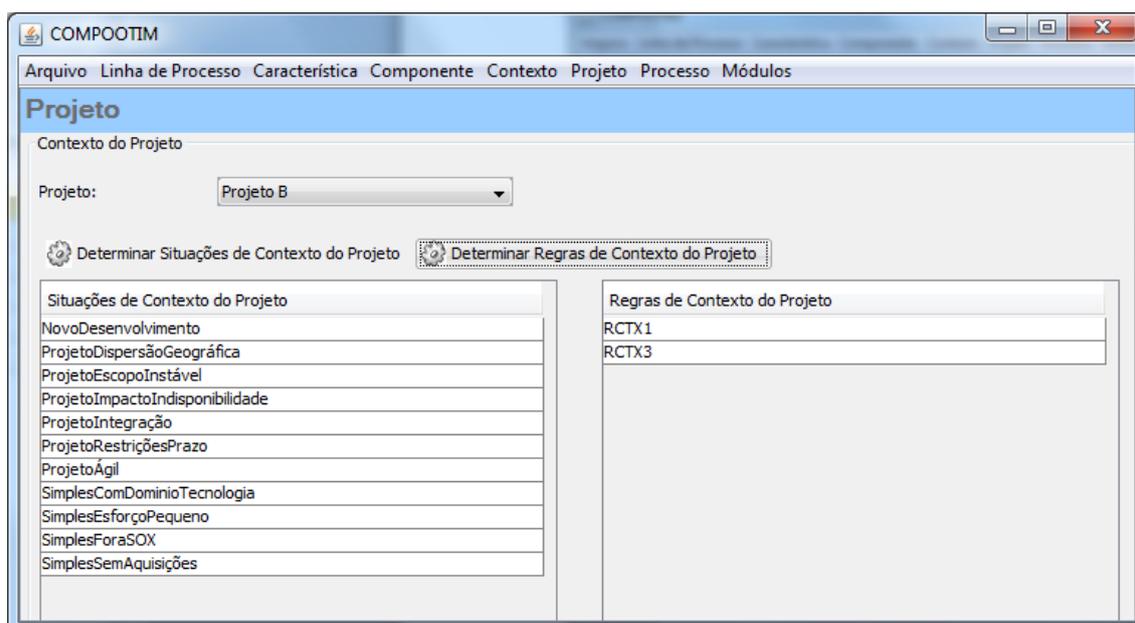


Figura 6.5 – Situações e Regras de Contexto do Projeto B na COMPOOTIM

Uma vez caracterizado o contexto de cada projeto, inicia-se o filtro de características. A COMPOOTIM permite que seja(m) selecionada(s) a(s) característica(s) opcional(is) que se deseja que esteja(m) presente no processo. De acordo com as características escolhidas, são inferidas com base nas regras de contexto as características do processo que serão recortadas da LPS. Em seguida, as regras de composição de características são aplicadas e também incluem ou excluem algumas características (Figura 6.6). A seleção dessas características também vai influenciar a seleção ou exclusão dos componentes de processo no filtro seguinte.

Para o Projeto A, foi selecionada apenas a característica “Levantar requisitos” e a aplicação dos filtros resultou em uma redução do universo de características de 86 para 59, sendo 21 mandatórias e 38 opcionais. No Projeto B, não foi assinalada nenhuma característica opcional para ser incluída no processo e os filtros restringiram o total de 86 características para 43, conforme ilustrado pela Figura 6.6.

COMPOOTIM

Arquivo Linha de Processo Característica Componente Contexto Projeto Processo Módulos

Composição do Processo

Filtrar Características

Projeto: Linha de Processo:

Marcar/Desmarcar Todos

Característica	Tipo	Opcionalidade	Variabilidade
<input type="checkbox"/> Acompanhar Projeto Ágil	Tarefa	Opcional	Variante
<input type="checkbox"/> Avaliar viabilidade de infraestrutura	Tarefa	Opcional	Invariante
<input type="checkbox"/> Backlog do produto	Produto	Opcional	Invariante
<input type="checkbox"/> Classificação do projeto	Produto	Mandatório	Invariante
<input type="checkbox"/> Classificar projeto de software	Tarefa	Mandatório	Invariante
<input type="checkbox"/> Construir solução de software ágil	Atividade	Opcional	Variante
<input type="checkbox"/> Detalhar iteração	Tarefa	Mandatório	Invariante
<input type="checkbox"/> Elaborar visão do projeto	Tarefa	Opcional	Invariante
<input type="checkbox"/> Encerramento	Atividade	Mandatório	Invariante
<input type="checkbox"/> Encerrar projeto	Tarefa	Mandatório	Invariante

Total

Total Antes do Filtro: Total Filtro Composição:

Tipo

Atividade: Tarefa: Produto:

Opcionalidade

Mandatório: Opcional:

Variabilidade

Ponto de Variação: Variante: Invariante:

Regras Atuentes

Regras de Composição:

Figura 6.6 – Filtro de Características do Projeto B na COMPOOTIM

No passo seguinte, no filtro de componentes de processo, também podem ser selecionados os componentes de processo que devem estar

presentes na otimização do processo. O filtro neste caso atua recortando os componentes de acordo com o recorte das características realizado no passo anterior e aplicando as regras de composição dos componentes.

Para o Projeto A foram selecionados 13 componentes opcionais, enquanto que para o Projeto B foram escolhidos apenas 6 componentes entre as opções disponíveis. Ainda usando o Projeto B como exemplo, na Figura 6.7 é possível observar que este filtro de componentes provocou uma redução do total de 86 componentes para 22, que foram em seguida submetidos à otimização.

Composição do Processo

Filtrar Componentes

Projeto: Linha de Processo:

Marcar/Desmarcar Todos

Componente	Característica	Opcionalidade	Variabilidade	Colaboração
<input type="checkbox"/> Avaliar resultado dos testes	Executar testes	Mandatório	Invariante	0.08
<input type="checkbox"/> Classificar projeto	Classificar projeto de software	Mandatório	Invariante	0.06
<input checked="" type="checkbox"/> Colocar em homologação	Solicitar implantação de software	Opcional	Invariante	0.0
<input type="checkbox"/> Colocar em produção	Solicitar implantação de software	Mandatório	Invariante	0.0
<input type="checkbox"/> Consolidar PGP	Solicitar aprovação de plano de projeto	Mandatório	Invariante	0.0
<input type="checkbox"/> Constituir equipe para execução do projeto	Planejar projeto de software	Mandatório	Invariante	0.11
<input checked="" type="checkbox"/> Designar papéis	Iniciar projeto	Opcional	Invariante	0.11
<input type="checkbox"/> Designar responsável pelo projeto	Classificar projeto de software	Mandatório	Invariante	0.11
<input type="checkbox"/> Detalhar orçamento do projeto	Planejar projeto de software	Mandatório	Invariante	0.0
<input type="checkbox"/> Elaborar cronograma do projeto	Planejar projeto de software	Mandatório	Invariante	0.14
<input type="checkbox"/> Elaborar declaração de escopo do projeto	Planejar projeto de software	Mandatório	Invariante	0.0

Total

Total Antes do Filtro: Opcionalidade: Mandatório: Variabilidade: Ponto de Variação:

Total Filtro: Opcional: Variante:

Invariante:

Regras Atuentes

Regras de Composição:

Figura 6.7 – Filtro de Componentes do Projeto B na COMPOOTIM

Após a execução da otimização (que será detalhada na Seção 6.5), as opções de processos resultantes foram apresentadas ao gestor do GDPS para avaliação. Esta etapa de avaliação foi realizada em uma única sessão com duração de 50 minutos.

Para o Projeto A, foram obtidas 576 opções de processo viáveis (Figura 6.8). Estas opções variam em relação ao potencial de colaboração alcançado, o número e a ordem de execução dos componentes e o número de artefatos que constituem o processo. O participante pôde navegar por uma seleção das

15 melhores entre estas opções, avaliando as diferenças entre elas e o processo descrito através da lista ordenada de componentes e artefatos.

Opções de Processos

Processo	Colaboração	# Componentes	# Artefatos
Processo 2	0.22	20	26
Processo 5	0.22	21	27
Processo 8	0.22	21	27
Processo 11	0.22	22	28
Processo 4	0.21	20	26

Figura 6.8 – Resultado da Otimização do Projeto B na COMPOOTIM

Para o Projeto B, foram obtidas 12 opções de processo. As diferenças entre estas opções são resumidas na Tabela 6.25, que destaca as similaridades e diferenças entre elas. Porém, vale ressaltar que esta tabela não foi utilizada no estudo de caso e que o participante também avaliou as opções do processo para o Projeto B diretamente na ferramenta COMPOOTIM.

Durante a avaliação dos processos resultantes, o gestor do GDPS fez algumas observações em relação aos componentes listados nos processos, conforme registrado na Tabela 6.26. Todos estes comentários dizem respeito aos dados dos componentes cadastrados e a novas necessidades de ajustes na LPS definida na etapa anterior do estudo. Porém, não foi detectado nenhum problema nos processos sugeridos.

Tabela 6.25 – Opções de Processo do Projeto B

Componente	1	2	3	4	5	6	7	8	9	10	11	12
Iniciar atendimento da demanda	X	X	X	X	X	X	X	X	X	X	X	X
Designar responsável pelo projeto	X	X	X	X	X	X	X	X	X	X	X	X
Classificar projeto	X	X	X	X	X	X	X	X	X	X	X	X
<i>Designar papéis</i>							X	X	X	X	X	X
Criar visão do projeto	X	X	X	X	X	X	X	X	X	X	X	X
Realizar reunião de planejamento da release	X	X	X	X	X	X	X	X	X	X	X	X
Realizar reunião de planejamento da sprint	X	X	X	X	X	X	X	X	X	X	X	X
<i>Iniciar execução de projeto</i>				X	X	X				X	X	X
Liberar versão	X	X	X	X	X	X	X	X	X	X	X	X
Implementar software	X	X	X	X	X	X	X	X	X	X	X	X
Gerenciar execução dos trabalhos do projeto	X	X	X	X	X	X	X	X	X	X	X	X
Realizar reunião diária	X	X	X	X	X	X	X	X	X	X	X	X
Solicitar passagem para homologação	X	X	X	X	X	X	X	X	X	X	X	X
Marcar linha-base de homologação	X	X	X	X	X	X	X	X	X	X	X	X
Colocar em homologação	X	X	X	X	X	X	X	X	X	X	X	X
<i>Demonstrar produto de software</i>		X	X		X	X		X	X		X	X
<i>Realizar retrospectiva da sprint</i>			X			X			X			X
Solicitar passagem para produção	X	X	X	X	X	X	X	X	X	X	X	X
Marcar linha-base de produção	X	X	X	X	X	X	X	X	X	X	X	X
Colocar em produção	X	X	X	X	X	X	X	X	X	X	X	X
Obter aceitação final	X	X	X	X	X	X	X	X	X	X	X	X
Encerrar projeto	X	X	X	X	X	X	X	X	X	X	X	X
TOTAL DE COMPONENTES	18	19	20	19	20	21	21	20	21	20	21	22
TOTAL DE ARTEFATOS	24	25	26	25	26	27	27	26	27	26	27	28

Tabela 6.26 – Comentários sobre as avaliações dos processos

Projeto	Componente	Comentários
Projeto A	Consolidar PGP	Escrever PGP por extenso (Plano de Gerenciamento de Projeto)
Projeto B	Classificar projeto	Não ocorre para projeto ágil - este componente não deve ser mandatório e sim opcional.
Projeto B	Demonstrar produto de software	Não fica claro no processo se é feito em homologação ou desenvolvimento, mas assumir que é no desenvolvimento e retirar os 3 componentes de instalação em homologação.
Projeto B	Realizar retrospectiva da sprint	Não fica claro no processo, mas pode ser antes ou depois da produção.

6.4.3. Análise dos Resultados

Esta seção resume os principais resultados obtidos com a execução da segunda etapa deste estudo de caso. Os resultados foram estruturados de acordo com as variáveis avaliadas. Algumas variáveis foram coletadas através do questionário preenchido pelo participante após a avaliação da LPS.

6.4.3.1. Viabilidade

Sobre a viabilidade da COMPOOTIM em compor um processo otimizado de acordo com o contexto do projeto, o participante afirmou que a COMPOOTIM conseguiu efetivamente compor os processos para os projetos e que *“isto foi verificado na ferramenta COMPOOTIM”*. O participante indicou ainda que ficou muito satisfeito com os resultados obtidos (Tabela 6.27), conforme indica também o baixo número de observações feitas sobre o processo resultante. (Tabela 6.26).

Sobre o tamanho e a completude dos processos apresentados como opção para os Projetos A e B em relação ao esperado pela organização, o participante informou que eles estão dentro do esperado (Tabela 6.27) e destacou como uma contribuição interessante da ferramenta: *“A COMPOOTIM nos dá opções de tamanho”*. Esta observação se refere ao fato da ferramenta permitir ordenar as opções de processo de acordo com o número de componentes e artefatos presentes em cada uma delas. Questionado ainda se os processos compostos respeitaram o contexto específico de cada projeto, o participante afirmou que sim.

Tabela 6.27 – Viabilidade

Item	Resposta
COMPOOTIM conseguiu efetivamente compor os processos	Sim
Grau de satisfação	Muito satisfeito
Tamanho relativo do processo do Projeto A	Esperado
Tamanho relativo do processo do Projeto B	Esperado
Completude do processo do Projeto A	Esperada
Completude do processo do Projeto B	Esperada
Contexto dos projetos	Sim

6.4.3.2. Vantagens, Desafios e Melhorias

Considerando-se a experiência do participante como gestor do GDPS, é possível imaginar que ele consiga julgar a viabilidade de uso prático da composição de processos na organização. O participante afirmou que a sua adoção prática parece viável e não fez ressalvas quanto a esta possibilidade de uso pela empresa, ou seja, não foram detectados empecilhos para este uso (Tabela 6.28).

Tabela 6.28 – Vantagens, Desafios e Melhorias

Item	Resposta
Composição de processos é viável para adoção prática	Adoção prática parece viável
Vantagens	Possibilidade de identificar o melhor cenário no processo para realizar um projeto de desenvolvimento. Ajuda até quem não tem muito conhecimento sobre Engenharia de Software
Desafios	Atualização das regras a partir das mudanças dos processos.
Aspectos Positivos	É importante ter uma ferramenta para auxiliar na composição.
Aspectos Negativos	Ainda pode ser trabalhada uma melhor forma de apresentar os resultados na ferramenta (questões de paralelismo de atividades e consolidação por assunto - projeto / desenvolvimento).
Sugestões de Melhoria	Além de utilizar as regras do processo para determinar a melhor forma de realizar um projeto, precisam ser pesados que outros fatores poderiam auxiliar na composição como, por exemplo, lições aprendidas e resultados dos projetos (base histórica).
Comentários Adicionais	Devo destacar a importância de elevar a maturidade da organização na definição da melhor solução de processo para seus projetos. A ferramenta auxilia, porém a necessidade de adquirir o conhecimento e a maturidade na realização dos projetos não podem ser desconsiderados. A ferramenta auxilia, mas não substitui o Engenheiro de Software.

A principal vantagem identificada pelo participante foi a *“possibilidade de identificar o melhor cenário no processo para realizar um projeto de desenvolvimento. Ajuda até quem não tem muito conhecimento sobre Engenharia de Software”*. Este comentário é um indício importante de que a COMPOOTIM atingiu os seus objetivos de apoiar um gerente de projeto durante a decisão de composição do processo.

Por sua vez, o principal desafio mencionado foi novamente a preocupação com a atualização das regras na criação da LPS e em como isso pode impactar os processos recortados para os projetos, que sofrem influência direta destas regras.

Como aspecto positivo, o participante destacou a importância de ter uma ferramenta para apoiar a composição de processos. Porém, como aspecto negativo, foi mencionada a necessidade de se melhorar a apresentação dos resultados da composição dos processos na ferramenta (Tabela 6.28). De fato, na visualização atual não ficam claras as possibilidades de paralelismo do

processo, principalmente no que diz respeito às atividades de gestão e desenvolvimento que no processo definido pela organização não possuem uma amarração clara. Assim, esta visualização poderia ser evoluída em trabalhos futuros que permitissem a sua representação visual.

O gestor do GDPS apresentou como sugestão de melhoria a possibilidade de se ponderar “*outros fatores que poderiam auxiliar na composição como, por exemplo, lições aprendidas e resultados dos projetos (base histórica)*”. Tal sugestão está alinhada com a visão do Projeto CDSOft de que outros fatores, além da colaboração explorada neste trabalho, podem ser otimizados na composição de processos no futuro.

Por fim, como comentários adicionais, o participante destacou a necessidade de “*eleva a maturidade da organização na definição da melhor solução de processo para seus projetos*”, pois reconhece que mesmo com o auxílio da ferramenta, continua existindo um responsável pelas decisões de composição do processo. Esta premissa de que o suporte computacional não substitui a atuação do gerente de projeto (ou de um líder responsável pela composição do processo) também está em conformidade com a visão defendida neste trabalho.

6.4.4. Conclusões e Limitações

Após a realização desta segunda etapa do estudo, pode-se afirmar que a sua principal contribuição foi ter avaliado, sem alterações significativas, as opções de processos sugeridas para os dois projetos de desenvolvimento, respeitando os seus respectivos contextos. Esta avaliação representa um indício importante da viabilidade de uso da sistemática e do ferramental de apoio de composição de processos. Além disso, a facilidade com que o gestor do GDPS conseguiu entender as sugestões de processos e avaliar a sua corretude, usando a ferramenta COMPOOTIM, também reforça os indícios de sua aplicabilidade prática.

Destaca-se também o fato do participante ter conseguido perceber o potencial da ferramenta em auxiliar até mesmo gerentes de projeto com menos experiência. Este é um indício importante de que a COMPOOTIM atingiu os seus objetivos de apoio durante a decisão de composição do processo. Ao mesmo tempo, foi reforçada a premissa de que o suporte computacional

contribui, mas não substitui a atuação do gerente de projeto.

Esta etapa do estudo se distinguiu da anterior pelo uso do apoio computacional e o próprio participante destacou a sua importância. Porém, melhorias na visualização dos resultados ainda são necessárias, principalmente na representação das sequências e paralelismos do processo. Uma possibilidade que pode ser explorada futuramente é adicionar um *plug-in* para a representação visual do processo.

Como principal limitação desta etapa do estudo, deve-se mencionar o número restrito de projetos usados nesta avaliação. Ainda que estes projetos tivessem características distintas, não é possível generalizar os resultados para outros tipos de projetos.

6.5. Avaliação da Otimização da Colaboração

Esta seção detalha a avaliação da otimização da colaboração. O insumo da otimização foi a lista de componentes filtrada pela COMPOOTIM para os projetos A e B. O Projeto A tinha 42 componentes, sendo 25 mandatórios, 15 opcionais e 2 inviáveis. O Projeto B tinha 22 componentes, sendo 14 mandatórios e 8 opcionais (Tabela 6.29).

Como o número de componentes opcionais, que variam nas sugestões de solução, do Projeto B era inferior a 10, para este caso foi utilizado o algoritmo de Força Bruta (FB). Para um espaço de busca tão pequeno, não é necessário pensar em aplicar um algoritmo de otimização. Esta estratégia de uso do FB pode ser aplicada sempre que a COMPOOTIM estiver lidando com processos pequenos, pois o tempo de execução do FB neste caso é desprezível. Para o Projeto B, este tempo foi inferior a 1 ms em todas as soluções. O FB foi capaz de encontrar as 12 alternativas de solução para o Projeto B e indicou que a solução com maior *fitness* atinge um nível de colaboração de 0.2259 utilizando a estratégia mediana.

No caso do Projeto A, que possuía um número de componentes opcionais para ser avaliado superior a 10, foi executada a otimização utilizando os algoritmos de *Hill Climbing* (HC), *Random Search* (RS) e FB. Os dois primeiros algoritmos receberam um limite máximo de avaliações correspondente ao número de componentes opcionais elevado ao quadrado. O

FB foi executado para avaliar se os dois primeiros foram capazes de encontrar as soluções ótimas para seus respectivos problemas.

Tabela 6.29 – Dados dos Projetos

	Projeto B	Projeto A
Total de Componentes	22	42
Componentes Mandatórios	14	25
Componentes Opcionais	8	15
Componentes Inviáveis	-	2
Algoritmos	FB (Tempo desprezível para processos pequenos)	HC, RS e FB
Limite de avaliações	-	n^2 ($n = n^o$ de componentes opcionais)
No de ciclos de execução	-	100
Alternativas de Solução	12	576
Solução com Maior <i>Fitness</i> (estratégia mediana)	0.2259	0.1484

A Tabela 6.30 sumariza os resultados da execução dos algoritmos HC, RS e FB para o Projeto A. Para o HC e o RS são apresentados o valor mínimo, a média, o desvio padrão e o valor máximo do *fitness* para a melhor solução encontrada ao longo dos 100 ciclos de execução. É importante frisar que a função de *fitness* está sendo maximizada pelos algoritmos.

Além dos valores do *fitness*, também foi calculado o *p-value* pelo teste estatístico não-paramétrico de Wilcoxon-Mann-Whitney (coluna PV) e o tamanho de efeito (coluna ES) para cada projeto. Estes valores foram calculados usando a ferramenta R Statistical Computing v3.0.1²⁶.

P-values próximos de zero indicam um alto nível de confiança de que os valores sendo comparados são estatisticamente diferentes. Na Tabela 6.30, o PV é significativamente pequeno, denotando que os resultados do HC são superiores aos resultados do RS com pelo menos 99% de certeza. Estes resultados mostram que o HC é mais eficaz do que o RS no sentido de encontrar melhores soluções para o problema com o valor máximo estabelecido para o número de avaliações da função de *fitness*.

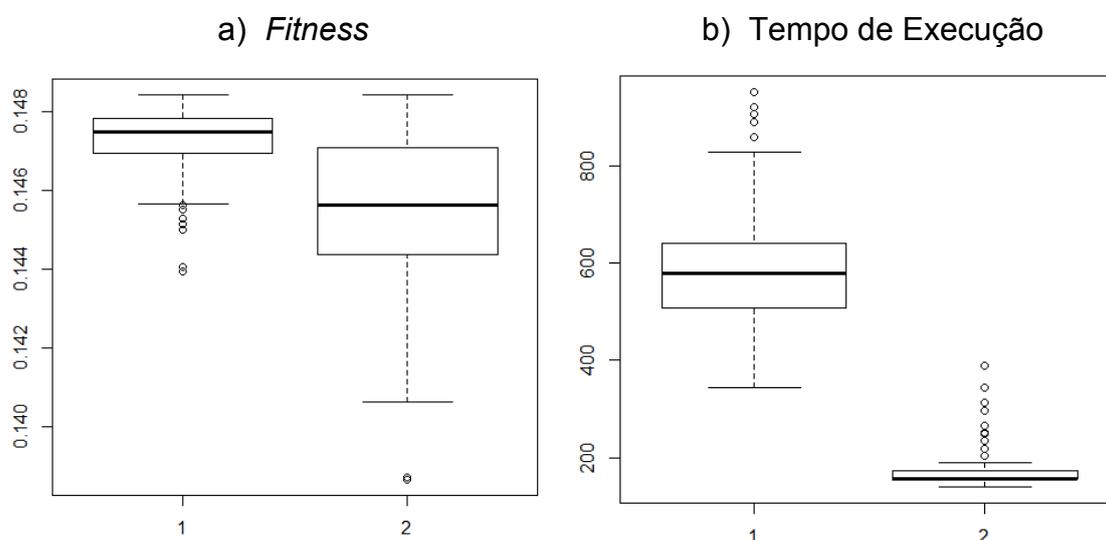
²⁶ Site: <http://www.r-project.org/>

Tabela 6.30 – Valores de *Fitness* dos algoritmos para o Projeto A

Min HC	HC	Max HC	Min RS	RS	Max RS	FB	PV	ES
0.1439	0.1472 ± 0.0009	0.1484	0.1386	0.1454 ± 0.0020	0.1484	0.1484	< 0.001	77%

Medidas de tamanho de efeito, como a não-paramétrica \hat{A}_{12} de Vargha e Delaney (ARCURI e BRIAND, 2011) utilizada neste estudo medem a magnitude da melhoria em uma comparação entre pares. Dada uma medida M para observações coletadas após a aplicação dos tratamentos A e B, \hat{A}_{12} mede a probabilidade do tratamento A atingir valores de M maiores que B. Se os dois tratamentos forem equivalentes, então $\hat{A}_{12} = 0.5$. Senão, a medida indica a frequência de melhoria do método A em relação ao método B. Por exemplo, $\hat{A}_{12} = 0.8$ denota que resultados melhores serão obtidos 80% das vezes com A. Neste estudo, conforme apresentado na Tabela 6.30, o HC produzirá melhores resultados em 77% das vezes no Projeto A.

Também é possível observar que o valor mínimo do HC é maior do que o valor mínimo encontrado pelo RS e que, na média, o HC alcança soluções com *fitness* maiores, ou seja, o HC é mais eficaz em encontrar soluções para a otimização da colaboração na composição do processo para o Projeto A. O gráfico de *box-plot* apresentado na Figura 6.9a confirma que o HC teve resultados de *fitness* superiores ao RS. Através do gráfico é possível perceber visualmente que, apesar de haver alguma sobreposição entre as amostras, claramente, o *fitness* do HC tende a encontrar soluções melhores.

Figura 6.9 – *Box-plot* do Projeto A

Os tempos de execução do Projeto A são apresentados na Tabela 6.31, que também mostra o valor mínimo, a média, o desvio padrão, o valor máximo, o PV e o ES de comparação entre o HC e o RS. O *p-value* do teste estatístico não-paramétrico Wilcoxon-Mann-Whitney está abaixo de 1% e o tamanho de efeito está em quase 100%, indicando que o HC consome mais tempo que o RS e que existe uma diferença significativa no tempo de execução destes. O gráfico de *box-plot* apresentado na Figura 6.9b confirma que os tempos de execução do HC foram maiores e variaram mais do que os tempos do RS.

A análise destes resultados permite concluir que o HC é mais eficaz (melhor resultado em relação à função de *fitness*) que uma busca aleatória, porém não é o mais eficiente (maior tempo de execução do algoritmo) na busca por soluções para este problema da otimização da colaboração na composição de processos de software.

Tabela 6.31 – Valores de Tempo de Execução do Projeto A (em ms)

Min HC	HC	Max HC	Min RS	RS	Max RS	PV	ES
344	590.27 ± 121.49	952	140	174.88 ± 43.32	390	< 0.001	99%

Na área de SBSE, apresentada no Capítulo 5, Clarke *et al.* (2003) sugerem que se comece pela técnica mais simples para verificar se os resultados são encorajadores antes de investir em técnicas mais sofisticadas, que podem exigir um esforço adicional desnecessário. Esta foi a estratégia seguida por este trabalho, que iniciou a investigação da otimização pelo RS e HC. Neste estudo, foi comparada a execução da técnica heurística (HC) com uma busca totalmente aleatória (RS). Como resultado, observou-se que a técnica heurística encontrou soluções melhores, porém com maior esforço computacional. A execução da otimização com uma massa de dados maior, que incluía um número maior de projetos, é necessária para confirmar os indícios providos por este estudo.

6.6. Considerações Finais

Neste capítulo, o estudo exploratório com especialistas em colaboração ajudou a avaliar a viabilidade de associar um componente de processo às práticas de

colaboração do CollabMM e indicou oportunidades de pesquisa futuras interessantes para dar continuidade às investigações sobre o tema, que também será explorado por (CLARET, 2013).

Em relação à EvolTrack-SocialNetwork, os resultados do estudo exploratório realizado indicam um *feedback* positivo sobre o potencial da ferramenta para aumentar a capacidade de percepção da colaboração em um projeto de software. O estudo também ajudou a identificar novos requisitos para melhorar a usabilidade da ferramenta.

O principal estudo experimental realizado nesta tese foi o estudo de caso que incluiu a avaliação da criação da LPS e da composição de processos. Para cada uma destas etapas, foram descritos os objetivos do estudo e seu planejamento e foi também apresentada a execução do estudo, descrevendo como cada etapa foi realizada e os resultados obtidos.

A primeira etapa do estudo forneceu indícios de que a sistemática é capaz de resultar em uma LPS completa e que faça sentido para o gestor do GDPS. A LPS criada poderá ser utilizada e expandida em futuras avaliações. A segunda etapa do estudo demonstrou a viabilidade da composição, otimização e visualização dos processos na ferramenta COMPOOTIM. O potencial da ferramenta em auxiliar até mesmo gerentes de projeto com menos experiência, é um indício importante de que a COMPOOTIM atingiu os seus objetivos de apoio durante a decisão de composição do processo.

Por fim, foi realizada uma avaliação dos resultados obtidos pelos diferentes algoritmos para a otimização da colaboração. Os resultados indicaram que o algoritmo HC é mais eficaz (melhor resultado em relação à função de *fitness*) que uma busca aleatória, porém não é o mais eficiente (maior tempo de execução do algoritmo) na busca por soluções para este problema da otimização da colaboração na composição de processos de software.

O próximo capítulo apresenta a conclusão desta tese.

7. Conclusão

Este capítulo enumera as conclusões obtidas com a realização do presente trabalho, especificando suas principais contribuições, discutindo as limitações e sugerindo perspectivas para trabalhos futuros.

7.1. Epílogo

Conforme apresentado ao longo desta tese, a colaboração é uma constante, embora apresente diversidade, no desenvolvimento de software. Apesar da sua importância para fins de qualidade, produtividade e compartilhamento de conhecimento, as organizações de software ainda têm dificuldades em planejar como a colaboração deve acontecer no seu dia-a-dia de trabalho e acompanhar os resultados alcançados.

Este trabalho investigou a possibilidade de planejar, acompanhar e otimizar a colaboração na composição de processos de software. A definição de um processo retrata como a interação entre os participantes é desejada e os dados da execução de um processo indicam como a colaboração aconteceu de fato. Assim, a colaboração é um aspecto passível de ser planejado e acompanhado nos processos definidos em distintos contextos de projetos de desenvolvimento de software.

Para definir processos de software de maneira mais efetiva é possível investir na reutilização de processos a partir de processos definidos previamente e na adequação ao contexto do projeto. Neste cenário de reutilização, a composição de processos, baseada em estruturas como componentes e linhas de processos, foi a estratégia adotada neste trabalho.

Para apoiar o gerente de projeto na composição de um processo de software que maximize a colaboração e seja aderente ao contexto do projeto em questão, foi proposta a COMPOOTIM. A COMPOOTIM é composta por diferentes soluções voltadas para o planejamento, composição, otimização e acompanhamento da colaboração em processos de software.

Entre as soluções desenvolvidas na COMPOOTIM é possível citar: i) uma estratégia de medição da colaboração em processos de software, com base no modelo de maturidade em colaboração CollabMM; ii) a sistemática que

define o passo-a-passo para a criação da uma LPS e para a composição dos processos a partir da LPS; iii) ferramental de apoio com mecanismos de composição, otimização e visualização; iv) a ferramenta EvoTrack-SocialNetwork de mineração, visualização e análise de redes sociais.

Alguns estudos experimentais foram conduzidos no contexto desta tese para avaliar os instrumentos propostos como parte da solução COMPOOTIM (Tabela 6.1).

Tabela 7.1 – Resumo das avaliações realizadas

Soluções da COMPOOTIM	Tipo de Avaliação	Indicações
Estratégia de Medição da Colaboração	Estudo Exploratório	<ul style="list-style-type: none"> • Viabilidade e aplicabilidade • Extensão de pesquisa
EvoTrack-SocialNetwork	Estudo Exploratório	<ul style="list-style-type: none"> • Acompanhamento da colaboração
Sistemática	Estudo de Caso	<ul style="list-style-type: none"> • Viabilidade de uso prático • LPS completa
Ferramental de Apoio	Estudo de Caso	<ul style="list-style-type: none"> • Conseguiu efetivamente compor os processos para os projetos • Conseguiu oferecer diferentes opções de tamanho • Opções de processos respeitaram o contexto dos projetos • Apoio aos gerentes com menos experiência

Um estudo exploratório visou avaliar a estratégia de medição da colaboração em relação à possibilidade de associar os componentes de processo às práticas de colaboração do CollabMM e calcular o potencial de colaboração desses componentes.

Outro estudo exploratório no ambiente acadêmico, usando projetos de software livre, buscou avaliar se é possível acompanhar a colaboração entre os membros da equipe de desenvolvimento, usando a ferramenta EvoTrack-SocialNetwork. Os resultados do estudo indicaram o potencial da ferramenta em oferecer percepção da colaboração em um projeto de software.

No estudo de caso com uma empresa de petróleo de grande porte no Rio de Janeiro, pretendia-se avaliar a aplicabilidade, viabilidade e facilidade de uso da COMPOOTIM. O estudo de caso foi realizado em duas etapas. Na primeira etapa, o objetivo era avaliar se a sistemática da COMPOOTIM é capaz de gerar uma LPS. Na segunda etapa, o objetivo era avaliar se a COMPOOTIM

era capaz de, a partir da LPS criada na etapa anterior, compor processos para dois projetos de desenvolvimento escolhidos, de acordo com o contexto desses projetos e de forma otimizada, usando o ferramental de apoio construído.

Além disso, para avaliar os processos sugeridos do ponto de vista da otimização da colaboração, também foi utilizada análise estatística para comparar os resultados obtidos com os diferentes algoritmos de otimização em relação ao *fitness* e ao tempo de execução.

Os resultados deste estudo de caso representam um indício importante da viabilidade e aplicabilidade das soluções propostas. A primeira etapa do estudo indicou que a sistemática é capaz de resultar em uma LPS completa e que faça sentido para o gestor do GDPS. A segunda etapa do estudo demonstrou a viabilidade da composição, otimização e visualização dos processos na COMPOOTIM. O potencial da ferramenta em auxiliar até mesmo gerentes de projeto com menos experiência é um indício importante de que a COMPOOTIM atingiu os seus objetivos de apoio durante a decisão de composição do processo.

7.2. Contribuições

Devido a sua amplitude e ao forte envolvimento com o Projeto CDSOft, esta tese de doutorado trouxe diversas contribuições, mas com pesos distintos. Para dar a devida importância a cada uma delas, estas contribuições foram separadas nas próximas seções.

7.2.1. Principais Contribuições

As principais contribuições desta tese são:

- As ideias e resultados parciais da presente tese de doutorado permitiram delinear o arcabouço geral do **Projeto CDSOft** (Capítulo 4) (WERNER *et al.*, 2011a) e definir o foco de atenção e escopo específicos desta tese neste contexto;
- Desenvolvimento da **sistemática** da COMPOOTIM de construção da LPS e composição de processos de software a partir da LPS, visando otimizar a colaboração (Seção 5.2.1) (MAGDALENO *et al.*, 2012a);
- **Modelagem formal do problema** da otimização na composição de processos de software (Seção 5.3.2) (MAGDALENO, 2010c, MAGDALENO

et al., 2010a);

- Desenvolvimento do mecanismo de otimização da COMPOOTIM, através da **implementação dos algoritmos** de Força Bruta, *Hill Climbing* e *Random Search* (Seção 5.3.3);
- Construção do **ferramental de apoio** da COMPOOTIM com os mecanismos de composição, otimização e visualização (Seção 5.5);
- Especificação e construção de uma ferramenta de mineração, visualização e análise de redes sociais (**EvoTrack-SocialNetwork**) (Seção 5.4.2) (VAHIA *et al.*, 2011);
- Criação de uma **LPS completa e real** a partir dos dados do estudo de caso da COMPOOTIM, que pode servir para outras avaliações futuras desta e de outras pesquisas relacionadas ao Projeto CDSOFT (Seção 6.3) e para o uso da própria empresa.

7.2.2. Contribuições Secundárias

As contribuições secundárias desta tese podem ser assim enumeradas:

1. Definição de uma estratégia inicial de medição da colaboração em processos de software, com base no CollabMM, que define como é calculado o potencial de colaboração de cada componente de processo e a colaboração do processo como um todo (Seção 5.1.2);
2. Identificação dos requisitos necessários para ferramentas de mineração, visualização e análise de redes sociais (Seção 5.4.2) e análise de algumas ferramentas existentes (MAGDALENO *et al.*, 2010c);
3. O plano e os resultados obtidos com o estudo exploratório da EvoTrack-SocialNetwork (Seção 6.2) (MAGDALENO *et al.*, 2012b);
4. O plano e os resultados obtidos com o estudo de caso da COMPOOTIM em um contexto real da indústria (Seções 6.3 e 6.4).

7.2.3. Resultados em Publicações e Orientações

Alguns dos resultados obtidos ao longo do desenvolvimento desta tese foram registrados nas seguintes publicações:

- MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L., 2012, “An Exploratory Study on Collaboration Understanding in Software Development Social Networks”, *International Conference on Collaboration and Technology (CRIWG)*, vol. 7493, Raesfeld, Germany: Springer, 2012, pp. 113–120.

- MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L., 2012, "COMPOOTIM: An Approach to Software Processes Composition and Optimization", *Congresso Ibero-Americano em Engenharia de Software (CIBSE)*, Buenos Aires, Argentina, 2012, pp. 1–14.
- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2012, "Reconciling Software Development Models: A Quasi-Systematic Review", *Journal of Systems and Software (JSS)*, vol. 85, no 2, p. 351–369, 2012.
- VAHIA, C. M.; MAGDALENO, A. M.; WERNER, C. M. L., 2011, "EvoTrack-SocialNetwork: Uma ferramenta de apoio à visualização de redes sociais". *Congresso Brasileiro de Software: Teoria e Prática (CBSOFT) – Sessão de Ferramentas*, pp. 7-13, São Paulo, SP, Brasil.
- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2011, "A Criação de um Exemplo para Alimentar a Abordagem COMPOOTIM de Composição e Otimização de Processos de Software", Relatório Técnico ES-744/11, PESC-COPPE.
- MAGDALENO, A. M. ARAUJO, R. M.; WERNER, C. M. L., 2011, "A Roadmap to the Collaboration Maturity Model (CollabMM) Evolution". *International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 105-112, Lausanne, Switzerland.
- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2011, *Um Roadmap para a Evolução do Modelo de Maturidade em Colaboração (CollabMM)*, Relatório Técnico ES-742/11, PESC-COPPE.
- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2010, "Analyzing Collaboration in Software Development Processes through Social Networks". In: *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, pp. 435-446, Heraklion, Crete, Greece.
- MAGDALENO, A. M.; BARROS, M.; WERNER, C. M. L.; ARAUJO, R. M., 2010, "Formulando a Adaptação de Processos de Desenvolvimento de Software como um Problema de Otimização". *I Workshop Brasileiro de Otimização em Engenharia de Software (WOES)*, pp. 56-64, Salvador, BA, Brasil.
- MAGDALENO, A. M., 2010, "An optimization-based approach to software development process tailoring". In: *PhD Track - International Symposium on Search Based Software Engineering (SSBSE)*, pp. 40-43, Benevento, Italy.
- MAGDALENO, A. M., 2010, "Balancing Collaboration and Discipline in Software Development Processes". In: *Doctoral Symposium of International Conference on Software Engineering (ICSE)*, pp. 331-332, Cape Town, South Africa.
- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2010, *Estudo de*

Ferramentas de Mineração, Visualização e Análise de Redes Sociais. Relatório Técnico ES-735/10, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.

- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2009, *Revisão Quasi-Sistemática da Literatura: Conciliação de processos de desenvolvimento de software*, Relatório Técnico ES-730, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.

Além dessas publicações, outros trabalhos realizados com os demais alunos do grupo de pesquisa resultaram em 7 publicações não diretamente relacionadas às contribuições desta tese:

- ARAUJO, R. M.; LEITE, A. M. S.; MAGDALENO, A. M.; WERNER, C. M. L., 2012, "Modelo de contexto para apoio à adaptação de processos de software com foco na colaboração", *iSys - Revista Brasileira de Sistemas de Informação* pp. 1-20.
- WERNER, C. M. L.; MURTA, L. G. P.; SCHOTS, M. *et al.*, 2011, "EvolTrack: A Plug-in-Based Infrastructure for Visualizing Software Evolution". *Workshop Brasileiro de Visualização de Software (WBVS)*, pp. 1-8, São Paulo, SP, Brasil.
- SANTOS, T. A. L. ARAUJO, R. M.; MAGDALENO, A. M., 2011, "Bringing Out Collaboration in Software Development Social Networks". *International Conference on Product Focused Software Development and Process Improvement - Short Papers*, pp. 18-21, Torre Canne, Italy.
- CEPEDA, R. D. S. V.; MAGDALENO, A. M.; MURTA, L. G. P.; WERNER, C. M. L., 2010, "EvolTrack: Improving Design Evolution Awareness in Software Development", *Journal of the Brazilian Computer Society (JBACS)*, Vol. 15, No. 2, pp. 117-131.
- SANTOS, T. A. L.; ARAUJO, R. M.; MAGDALENO, A. M., 2010, "Identifying Collaboration Patterns in Software Development Social Networks", *Infocomp - Journal of Computer Science - Special Issue*, pp. 51-60.
- NUNES, V. T.; MAGDALENO, A. M.; WERNER, C. M. L., 2010, *Modelagem de contexto sobre o domínio de processos de desenvolvimento de software*. Relatório Técnico ES-734/10, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.
- SANTOS, T. A. L.; ARAUJO, R. M.; MAGDALENO, A. M., 2009, "Padrões para Percepção da Colaboração em Redes Sociais de Desenvolvimento de Software". *Workshop de Desenvolvimento Distribuído de Software (WDDS)*, pp. 1-10, Fortaleza, CE, Brasil.

Por fim, vale destacar a participação da autora desta tese como coorientadora em projetos de iniciação científica (VAHIA *et al.*, 2011) em tema relacionado a este trabalho e também a participação auxiliar na pesquisa de

mestrado de Santos (2010), Teixeira (2011) e Leite (2011). Essas pesquisas foram desenvolvidas com base nas soluções propostas nesta tese e foram uma oportunidade de aplicá-las.

7.3. Limitações

Esta seção destaca algumas limitações da solução desenvolvida nesta tese.

Uma estrutura simplificada para a definição e representação dos componentes de processos foi adotada na COMPOOTIM. Esta simplificação foi necessária para contornar a ausência, no meta-modelo e notação *Odyssey-ProcessFEX* (TEIXEIRA, 2011), adotada no Projeto CDSOFT e nesta tese, do conceito de componentes de processo e do sequenciamento/fluxo dos processos, necessários para a composição e otimização dos processos. Uma revisão deste meta-modelo está em andamento no escopo de outra tese de doutorado.

Como limitação do CollabMM, vale salientar que este modelo de maturidade em colaboração foi originalmente proposto para ser aplicado em processos de negócio em geral, sem estar voltado para um domínio específico. A particularização do modelo para o domínio de processos de software não foi alvo desta pesquisa. A estratégia de medição da colaboração proposta para a COMPOOTIM e baseada no CollabMM torna-se, portanto, genérica, desconsiderando possíveis características de colaboração específicas para o domínio de processos de software.

A estratégia de medição da colaboração proposta no Capítulo 5 incluía o cálculo do potencial de colaboração dos componentes e a atribuição do grau de colaboração do processo como um todo. A primeira parte foi avaliada através de um estudo exploratório que demonstrou a viabilidade da associação entre os componentes de processo e as práticas de colaboração. Porém, ainda precisa ser explorada a viabilidade de uso de cada uma das alternativas de cálculo do grau de colaboração do processo. Estas alternativas não foram comparadas durante a realização do estudo de caso.

Apesar das características de colaboração em redes sociais, voltadas para o aspecto da coordenação, serem parte integrante da solução deste trabalho para o acompanhamento e entendimento da colaboração, estas

características não foram implementadas na ferramenta EvolTrack-SocialNetwork e, portanto, as redes sociais ainda não são automaticamente classificadas de acordo com estas características.

Ainda como limitação da EvolTrack-SocialNetwork, é importante destacar que hoje a ferramenta trabalha com repositórios de gerência de configuração como fonte de dados. A arquitetura da ferramenta, herdada da EvolTrack e projetada para ser flexível, permite a inclusão futura de outras fontes de dados. Em particular, o uso de uma fonte de dados baseada em processos, como o log de execução de uma ferramenta de BPMS (*Business Process Management System*), seria importante para permitir de fato o acompanhamento da colaboração nos processos.

Em relação aos mecanismos de otimização implementados na COMPOOTIM, pode ser considerada uma limitação o fato de os algoritmos terem sido avaliados com base em um número restrito de projetos. O uso de uma massa de dados com um número maior de projetos seria importante para que fosse possível obter conclusões definitivas sobre a adequação destes algoritmos para o problema em questão ou sobre a necessidade de implementação de algoritmos mais sofisticados.

Sobre a infraestrutura computacional geral das soluções propostas nesta tese, vale ressaltar que apesar de ter sido originalmente concebida de forma integrada, ainda não foram criadas as interfaces entre os ambientes previstos nesta infraestrutura. Então, dados vindos do Odyssey, por exemplo, por enquanto, são alimentados manualmente na COMPOOTIM. Por sua vez, a EvolTrack-SocialNetwork ainda não possui integração com os demais ambientes.

Por fim, é possível destacar como limitação do trabalho o fato de o estudo de caso (descrito no Capítulo 6) ter sido realizado em estágio avançado da tese. Assim, não pôde contribuir efetivamente para a evolução das propostas do trabalho, tendo foco maior em avaliar ou confirmar as contribuições desta tese.

7.4. Perspectivas Futuras

O trabalho realizado nesta tese pode ser considerado um passo importante

para concretização dos resultados do Projeto CDSOft. Uma vez que ainda há trabalhos a serem realizados tanto para avaliar de maneira mais detalhada as soluções construídas quanto para evoluí-las, as perspectivas de trabalhos futuros assim se destacam:

a) Melhorias das Soluções Propostas

Durante a realização do estudo exploratório, foram identificadas possíveis melhorias para a ferramenta EvolTrack-SocialNetwork. Com base nas propriedades de ARS calculadas, a ferramenta já poderia indicar no painel de análise a característica de colaboração mais representativa para cada versão da rede apresentada. Além disso, também foram observadas oportunidades de melhorias relacionadas à escalabilidade de apresentação dos diagramas das redes. Estas funcionalidades podem ser incorporadas à EvolTrack-SocialNetwork.

O estudo de caso realizado apontou a necessidade de rever a sistemática em relação à necessidade das regras de composição, pois algumas delas parecem óbvias e poderiam ser substituídas por mecanismos de sequenciamento dos componentes de processo. Após esta revisão, também deve ser investigada a necessidade de todas as etapas e artefatos previstos na sistemática da COMPOOTIM, visando avaliar se esta sistemática pode ser adaptada ou simplificada de acordo com as necessidades de uso e representação de cada organização.

Também pode ser aperfeiçoada a visualização das opções de processo no ferramental de apoio da COMPOOTIM, principalmente na representação das sequências e paralelismos do processo. Isto pode ser feito adicionando uma representação gráfica.

b) Refinamento das Avaliações

O interesse da empresa onde foi realizado o estudo em manter um esforço de pesquisa conjunto representa uma oportunidade concreta para a ampliação futura da LPS criada e para a realização de novas avaliações. Assim, de forma a complementar as avaliações realizadas nesta tese, poderia ser repetido o estudo experimental com um universo maior de projetos, de modo a aumentar a significância dos resultados obtidos e reduzir as ameaças ao estudo.

No presente trabalho de pesquisa, optou-se por não fixar uma única forma de cálculo do nível de colaboração do processo. Como a medição da colaboração ainda é uma questão de pesquisa em aberto, optou-se por oferecer maior flexibilidade que permita a escolha da estratégia de medição mais adequada em cada contexto. Assim, foram disponibilizadas quatro estratégias alternativas, cuja escolha é feita pelo gerente de projeto como parte do planejamento da colaboração. Assim, podem ser planejadas e conduzidas avaliações que busquem validar estas estratégias, explorando as diferentes alternativas propostas.

Por fim, seria interessante conduzir avaliações que busquem verificar, mais precisamente, a eficácia e eficiência dos algoritmos de otimização implementados na COMPOOTIM, visando identificar a necessidade de inclusão de outros algoritmos. A execução da otimização com uma massa de dados maior, que inclua um número maior de projetos, é necessária para confirmar os indícios providos pelo estudo realizado.

Como trabalhos futuros de investigação do uso destes algoritmos, é possível avaliar o HC, comparando-o com outra técnica (como os algoritmos genéticos) que implemente a solução do mesmo problema para analisar o tempo e a qualidade da solução.

A análise de sensibilidade é outra opção de estratégia de avaliação. A análise de sensibilidade permite investigar o grau de influência que cada um dos parâmetros de entrada provoca na solução e identificar quais parâmetros contribuem mais para a localização da solução ótima. Esta análise consiste em selecionar cada um dos parâmetros, variar o valor do mesmo mantendo os valores dos demais fixos e observar a influência na qualidade da solução gerada e no tempo de resposta (HARMAN, 2007, NETTO, 2010).

c) Extensão da Pesquisa

Como oportunidades de extensão da pesquisa é possível avançar os trabalhos atuais do Projeto CDSOFT, de forma que os processos compostos a partir da LPS possam ser executados, por exemplo, em um mecanismo de workflow ou BPMS (*Business Process Management System*) (ARAUJO e BORGES, 2001, MAGDALENO *et al.*, 2007b, PUNTAR *et al.*, 2009) para automatizar a execução do processo. Uma alternativa é o uso da Charon (MURTA, 2002),

construída dentro do ambiente Odyssey e que oferece recursos para a modelagem, instanciação, simulação, execução, acompanhamento e evolução de processos de software.

A partir daí, a sistemática da COMPOOTIM pode ser estendida para que ao final da execução do processo, a organização ou o gerente de projeto analise a adequação do processo e, possivelmente, opte em realimentar a LPS, com novas opcionalidades ou variabilidades para aumentar o reuso em projetos futuros.

Outra possibilidade é aumentar as possibilidades de tomada de decisão do gerente do projeto durante a composição de processo, permitindo que ele escolha um processo entre as opções indicadas, com base em: processos similares ao que ele próprio costuma escolher em outros casos; análise das avaliações/comentários sobre cada um dos processos que foram incluídos por quem já o executou; recomendação social baseada no conhecimento de que outros gerentes de projeto costumam escolher cada um dos processos; e, finalmente, com base em uma indicação do sistema que analise o perfil deste gerente de projeto.

Uma oportunidade de pesquisa interessante é buscar meios de associar as informações de contexto do projeto ao planejamento do nível de colaboração necessário para o projeto. Assim, seria possível determinar em que situações a colaboração é bem-vinda no projeto. A partir do contexto do projeto também seria possível definir o nível de maturidade em colaboração que este projeto deveria buscar alcançar. Para isso, é preciso relacionar as informações de contexto com as características de colaboração dos níveis de maturidade do CollabMM.

Outro caminho possível é evoluir as pesquisas sobre medição da colaboração, visto que este ainda é um problema em aberto na área. Partindo da estratégia de medição proposta nesta tese, é possível atribuir pesos distintos que determinem o quanto um determinado componente contribui positiva ou negativamente para uma determinada prática de colaboração.

Potencialmente, um componente pode contribuir para várias práticas de colaboração, mas durante sua execução isso pode não acontecer. A estratégia de medição do PC do componente, de fato, é uma medida atribuída durante o planejamento da colaboração. Porém, este PC pode não se confirmar durante

a execução do processo. Então, durante o acompanhamento da colaboração, existe a necessidade de avaliar ao longo do tempo os valores de PC assumidos para cada componente.

Também é possível aproximar as propriedades de análise de redes sociais, adotadas na EvoTrack-SocialNetwork, das informações da otimização da colaboração de forma que o planejamento e o acompanhamento da colaboração se encaixem melhor e permitam um gestão efetiva da colaboração nos processos de desenvolvimento de software.

Neste sentido, deve-se buscar a integração entre os ambientes construídos no Projeto CDSOft como parte da solução proposta para a composição de processos. Esta ação inclui a integração entre o Odyssey, a COMPOOTIM, a infraestrutura de Gestão de Contexto que está em construção e a EvoTrack-SocialNetwork. Além disso, também almeja-se que o ambiente integrado construído evolua para se tornar um ambiente de apoio à decisão durante a composição de processos de software.

Por fim, outros aspectos de interesse, além da colaboração, podem ser explorados no contexto do Projeto CDSOft. Um deles, como a necessidade de disciplina/controle/formalismo pode funcionar até mesmo como um contraponto e ser balanceado com a colaboração (MAGDALENO, 2010a).

Referências Bibliográficas

- AALST, W. M. P.; GIINTHER, C. W., 2007, "Finding Structure in Unstructured Processes: The Case for Process Mining". *Application of Concurrency to System Design (ACSD)*, p. 3–12, Bratislava, Slovakia.
- AALST, W.; REIJERS, H. A.; SONG, M., 2005, "Discovering Social Networks from Event Logs", *Computer Supported Cooperative Work (CSCW)*, v. 14, n. 6 (jan.), p. 549–593.
- AALST, W. VAN DER; WEIJTERS, T.; MARUSTER, L., 2004, "Workflow Mining: Discovering Process Models from Event Logs", *IEEE Transaction on Knowledge and Data Engineering*, v. 16, n. 9, p. 1128–1142.
- ABAD, Z. S. H.; SADI, M. H.; RAMSIN, R., 2010, "Towards Tool Support for Situational Engineering of Agile Methodologies". In: *17th Asia Pacific Software Engineering Conference (APSEC)*, p. 326 –335, Sydney, Australia.
- ABRAHAMSSON, P.; WARSTA, J.; SIPONEN, M. T.; *et al.*, 2003, "New directions on agile methods: a comparative analysis". In: *Proceedings of the 25th International Conference on Software Engineering*, p. 244–254, Portland, Oregon, USA.
- ABRANTES, J. F.; TRAVASSOS, G. H., 2007, *Revisão quasi-Sistemática da Literatura: Caracterização de Métodos Ágeis de Desenvolvimento de Software*, Relatório Técnico ES-714/07, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.
- ACUNA, S. T.; DE ANTONIO, A.; FERRE, X.; *et al.*, 2000, "The Software Process: Modelling, Evaluation and Improvement", *Handbook of Software Engineering and Knowledge Engineering*, p. 1–35.
- ALBA, E.; CHICANO, J. F., 2007, "Software project management with GAs", *Inf. Sci.*, v. 177, n. 11 (jun.), p. 2380–2401.
- ALBERT, R.; BARABASI, A. L., 2002, "Statistical mechanisms of complex networks", *Reviews of Modern Physics*, v. 74, n. 1, p. 47–97.
- ALEIXO, A. I. DE S., 2003, *Procedimentos para Implantar Equipes*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia de

Produção, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil. Disponível em: <http://teses.eps.ufsc.br/defesa/pdf/8183.pdf>.

- ALEIXO, F. A.; FREIRE, M. A.; SANTOS, W. C.; *et al.*, 2010, "Uma Abordagem para Gerência e Customização de Variabilidades em Processos de Software". *Simpósio Brasileiro de Engenharia de Software (SBES) - Congresso Brasileiro de Software: Teoria e Prática (CBSOFT)*, p. 119–128, Salvador, BA, Brasil.
- ALONSO, J.; DE SORIA, I. M.; ORUE-ECHEVARRIA, L.; *et al.*, 2010, "Enterprise Collaboration Maturity Model (ECMM): Preliminary Definition and Future Challenges", *Enterprise Interoperability IV*, London: Springer, p. 429–438.
- ALTMANN, J.; POMBERGER, G., 1999, "Cooperative software development: concepts, model and tools". *Technology of Object-Oriented Languages and Systems (TOOLS)*, p. 194–207, Santa Barbara, CA, USA.
- AMBLER, S. W., 2008. Agile CMMI: Complimentary or Oxymoronic? *Dr. Dobb's*. Disponível em: <http://www.drdoobbs.com/architecture-and-design/212501388;jsessionid=RRCW3ZPM5BKMLQE1GHOSKH4ATMY32JVN?cid=Ambysoft>. Acesso em: 7 nov 2010.
- ANASTASSIU, M., 2012, *Um Método para Identificação de Elementos Contextuais que Impactam em Processos de Negócio*. Dissertação de Mestrado, PPGI/UNIRIO, Rio de Janeiro, RJ, Brasil.
- ANDERSON, D. J., 2010, *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- ARAUJO, R., 2000, *Ampliando a Cultura de Processos de Software: Um Enfoque Baseado em Groupware e Workflow*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- ARAUJO, R.; BORGES, M. R. S., 2001, "Sistemas de Workflow". In: *Jornadas de Atualização em Informática (JAI) – Congresso da Sociedade Brasileira de Computação (SBC)*, p. 1–17, Fortaleza, Ceará, Brasil.
- ARAUJO, R. M. DE; BORGES, M. R. S., 2007, "The role of collaborative support to promote participation and commitment in software development teams", *Software Process: Improvement and Practice*, v. 12, n. 3, p. 229–246.

- ARAUJO, R. M. DE; LEITE, A. M. S.; MAGDALENO, A. M.; *et al.*, 2012, "Modelo de contexto para apoio à adaptação de processos de software com foco na colaboração", *iSys - Revista Brasileira de Sistemas de Informação*, p. 1–20.
- ARAUJO, R. M. DE; SANTORO, F. M.; BRÉZILLON, P.; *et al.*, 2004, "Context Models for Managing Collaborative Software Development Knowledge". In: *Workshop on Modeling and Retrieval of Context (MRC)*, p. 61–72, Ulm.
- ARAUJO, R. M.; SANTORO, F. M.; BORGES, M. R. S., 2002, "The CSCW Lab for Groupware Evaluation", In: Haake, J. M., Pino, J. A. [orgs.] (eds), *Groupware: Design, Implementation, and Use*, , chapter 2440, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 222–231.
- ARCURI, A.; BRIAND, L., 2011, "A practical guide for using statistical tests to assess randomized algorithms in software engineering". In: *Proceedings of the 33rd International Conference on Software Engineering*, p. 1–10, New York, NY, USA.
- ARIMOTO, M. M.; MURAKAMI, E.; CAMARGO, V. V.; *et al.*, 2009, "Adherence Analysis of Agile Methods According to MR-MPS Reference Model". *Simpósio Brasileiro de Qualidade de Software (SBQS)*, p. 249–263, Ouro Preto, MG, Brasil.
- ARMBRUST, O.; KATAHIRA, M.; MIYAMOTO, Y.; *et al.*, 2008, "Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards", *Making Globally Distributed Software Development a Success Story*, Berlin / Heidelberg: Springer, p. 160–172.
- BALIEIRO, M. A.; JÚNIOR, S. F. S.; SOUZA, C. R. B., 2008, "Facilitating Social Network Studies of FLOSS using the OSSNetwork Environment", *Open Source Development, Communities and Quality*, , chapter 275, Boston: Springer, p. 343–350.
- BARABASI, A. L., 2003, *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Cambridge, Plume.
- BARABASI, A. L.; BONABEAU, E., 2003, "Scale-Free Networks", *Scientific American*, p. 50–59.

- BARBIERI, C., 2010. Mapeamento entre MPS.BR e Scrum-Introdução. *Blog do Barbi-Carlos Barbieri*. Disponível em: <http://blogdobarbi.blogspot.com/>. Acesso em: 6 nov 2010.
- BARNETT, L., 2004, "Applying Open Source Processes In Corporate Development Organizations", *Forrester Research*, p. 1–15.
- BARRETO, A., 2007, *Uma Abordagem para Definição de Processos de Software Baseada em Reutilização*. Exame de Qualificação, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BARRETO, A.; MURTA, L. G. P.; ROCHA, A. R., 2009, "Componentizando Processos Legados de Software Visando a Reutilização de Processos". *Simpósio Brasileiro de Qualidade de Software (SBQS)*, p. 189–203, Ouro Preto, Minas Gerais, Brasil.
- BARRETO, A.; MURTA, L. G. P.; DA ROCHA, A. R., 2007, "Uma Abordagem de Definição de Processos de Software Baseada em Reutilização". In: *ProQuality (UFLA)III Workshop de Implementadores MPS.BR*, p. 33–39, Campinas, SP, Brasil.
- BARRETO, A.; ROCHA, A. R.; MURTA, L., 2010, "Supporting the Definition of Software Processes at Consulting Organizations via Software Process Lines". In: *International Conference on the Quality of Information and Communications Technology (QUATIC)*, p. 15–24, Porto, Portugal.
- BARRETO, A. S., 2011, *Uma Abordagem para Definição de Processos baseada em Reutilização Visando à Alta Maturidade em Processos*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BARRETO, A. S.; MURTA, L. G. P.; CAVALCANTI DA ROCHA, A. R., 2011, "Software Process Definition: a Reuse-based Approach", *Journal of Universal Computer Science (JUICS)*, v. 17, n. 13, p. 1765–1799.
- BARTHELMESS, P., 2003, "Collaboration and coordination in process-centered software development environments: a review of the literature", *Information and Software Technology*, v. 45, n. 13 (out.), p. 911–928.
- BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D., 1994, "The goal question metric approach", *Encyclopedia of software engineering*, v. 1, p. 528–532.

- BASSI, D. L., 2008, *Experiências com Desenvolvimento Ágil*. Dissertação de Mestrado, Universidade de São Paulo (USP), São Paulo, SP, Brasil. Disponível em: <http://www.teses.usp.br/teses/disponiveis/45/45134/tde-06072008-203515/pt-br.php>.
- BECK, K., 2004, *Extreme Programming Explained: Embrace Change*. 2 ed. Boston, MA, USA, Addison-Wesley.
- BECK, K.; BEEDLE, M.; BENNEKUM, A. VAN; *et al.*, 2001. Manifesto for Agile Software Development. Disponível em: <http://agilemanifesto.org/>. Acesso em: 15 dez 2008.
- BECKER, J.; KNACKSTEDT, R.; PÖPPELBUß, J., 2009, "Developing Maturity Models for IT Management", *Business & Information Systems Engineering*, v. 1, n. 3 (maio.), p. 213–222.
- BEKKERS, W.; VAN DE WEERD, I.; BRINKKEMPER, S.; *et al.*, 2008, "The Influence of Situational Factors in Software Product Management: An Empirical Study". In: *International Workshop on Software Product Management (IWSPM)*, p. 41–48, Barcelona, Catalonia, Spain.
- BERGER, P. M., 2003, *Instanciação de Processos de Software em Ambientes Configurados na Estação TABA*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BETTIO, K.; VALASKI, J.; GOMES, D. L.; *et al.*, 2011, "Uma Experiência de Implementação Nível G em uma Empresa de Software Livre". *Simpósio Brasileiro de Qualidade de Software (SBQS)*, p. 409–416, Curitiba, PR, Brasil.
- BIOLCHINI, J.; MIAN, P. G.; NATALI, A. C. C.; *et al.*, 2005, *Systematic Review in Software Engineering*, Relatório Técnico ES-679, PESC-UFRJ. Disponível em: <http://www.cos.ufrj.br>.
- BIRD, C.; BARR, E.; NASH, A.; *et al.*, 2009a, "Structure and Dynamics of Research Collaboration in Computer Science". *SIAM International Conference on Data Mining (SDM)*, p. 826–837, Sparks, Nevada, USA.
- BIRD, C.; NAGAPPAN, N.; GALL, H.; *et al.*, 2009b, "Putting It All Together: Using Socio-technical Networks to Predict Failures". In: *Software Reliability Engineering, International Symposium on*, p. 109–119, Los Alamitos, CA, USA.

- BLOIS, A. P. T. B., 2006, *Uma Abordagem de Projeto Arquitetural Baseado em Componentes no Contexto de Engenharia de Domínio*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- BOEHM, B.; TURNER, R., 2003, *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston, MA, USA, Addison-Wesley.
- BORRELLI, G.; CABLE, J.; HIGGS, M., 1995, "What makes teams work better", *Team Performance Management*, v. 1, n. 3, p. 28–34.
- BOSCH, J., 2004, "Software Variability Management". In: *International Conference on Software Engineering (ICSE)*, p. 720–721, Scotland, UK.
- BOSCH, J.; BOSCH-SIJTSEMA, P. M., 2010, "Softwares Product Lines, Global Development and Ecosystems: Collaboration in Software Engineering", In: Mistrík, I., Grundy, J., Hoek, A., *et al.* [orgs.] (eds), *Collaborative Software Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 77–92.
- BOUGHZALA, I.; DE VREEDE, G.-J., 2012, "A Collaboration Maturity Model: Development and Exploratory Application". *45th Hawaii International Conference on System Science (HICSS)*, p. 306–315, Manoa, Hawaii, USA.
- BOYD, D.; ELLISON, N., 2007, "Social network sites: Definition, history, and scholarship", *Journal of Computer-Mediated Communication*, v. 13, n. 1, p. 1–11.
- BRANDES, U.; WAGNER, D., 2003, "Visone - Analysis and visualization of social networks", *Graph Drawing Software*, Heidelberg: Springer-Verlag, p. 321–340.
- BREZILLON, P., 1999, "Context in problem solving: a survey", *Knowledge Engineering Review*, v. 14, n. 1, p. 47–80.
- BRINKKEMPER, S., 1996, "Method engineering: engineering of information systems development methods and tools", *Information and Software Technology (IST)*, v. 38, n. 4, p. 275–280.
- DE BRUIN, T.; ROSEMAN, M.; FREEZE, R.; *et al.*, 2005, "Understanding the main phases of developing a maturity assessment model". In: *Australasian Conference on Information Systems (ACIS)*, p. 1–11, Sydney, Australia.

- BURNS, R. N.; DENNIS, A. R., 1985, "Selecting the appropriate application development methodology", *SIGMIS Database*, v. 17, n. 1, p. 19–23.
- CAPILUPPI, A.; LAGO, P.; MORISIO, M., 2003, "Characteristics of Open Source Projects". In: *Proceedings of the 17th European Conference on Software Maintenance and Reengineering*, p. 1–11, Washington, USA.
- CARVER, J.; JACCHERI, L.; MORASCA, S.; *et al.*, 2003, "Issues in using students in empirical studies in software engineering education". *International Software Metrics Symposium*, p. 239– 249, Sydney, Australia.
- CATALDO, M.; HERBSLEB, J. D., 2008, "Communication networks in geographically distributed software development". In: *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, p. 579–588, San Diego, CA, USA.
- CATALDO, M.; HERBSLEB, J. D.; CARLEY, K. M., 2008, "Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity". In: *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, p. 2–11, New York, NY, USA.
- CATALDO, M.; WAGSTROM, P. A.; HERBSLEB, J. D.; *et al.*, 2006, "Identification of coordination requirements: implications for the Design of collaboration and awareness tools". In: *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, p. 353–362, New York, NY, USA.
- CEPEDA, R. DA S. V.; MURTA, L. G. P.; WERNER, C., 2008, "Visualizando a Evolução de Software no Desenvolvimento Distribuído". *Workshop de Desenvolvimento Distribuído de Software (WDDS) - Simpósio Brasileiro de Engenharia de Software (SBES)*, p. 41–50, Campinas, SP, Brasil.
- CHAU, T.; MAURER, F.; MELNIK, G., 2003, "Knowledge sharing: agile methods vs. Tayloristic methods". *Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, p. 302–307, Linz, Austria.
- CHRISSIS, M. B.; KONRAD, M.; SHRUM, S., 2006, *CMMI: Guidelines for Process Integration and Product Improvement*. 2 ed. Boston, USA, Addison-Wesley.

- CLARET, M. D., 2013, "Métricas para Colaboração em Processos de Negócio". *Workshop de Teses e Dissertações - Simpósio Brasileiro de Sistemas de Informação (SBSI)*, João Pessoa, PB, Brasil (to appear).
- CLARKE, J.; DOLADO, J. J.; HARMAN, M.; *et al.*, 2003, "Reformulating software engineering as a search problem", *IEEE Proceedings-Software*, v. 150, n. 3, p. 161–175.
- COCKBURN, A., 2000, "Selecting a Project's Methodology", *IEEE Softw.*, v. 17, n. 4, p. 64–71.
- COCKBURN, A., 2001, *Agile Software Development*. Boston, MA, USA, Addison-Wesley.
- COCKBURN, A., 2004, *Crystal Clear: A Human-Powered Methodology for Small Teams*. 1 ed. Boston, MA, USA, Addison-Wesley.
- COCKBURN, A.; HIGHSMITH, J., 2001, "Agile Software Development: The People Factor", *IEEE Computer*, v. 34, n. 11, p. 131–133.
- COHAN, S.; GLAZER, H., 2009, "An Agile Development Team's Quest for CMMI Maturity Level 5". *Agile Conference (AGILE)*, p. 201–206, Chicago, USA.
- CONRADI, R.; FERNSTRÖM, C.; FUGGETTA, A., 1994, "Software process modelling and technology", In: Finkelsteiin, A., Kramer, J., Nuseibeh, B. [orgs.] (eds) Taunton, UK: Research Studies Press Ltd., p. 9–31.
- COSTA, A. J. S., 2010, *Um Mecanismo de Adaptação de Processos de Software*. Dissertação de Mestrado, Universidade Federal do Pará (UFPA), Belém, PA, Brasil.
- COSTA, J.; FEITOSA, R.; DE SOUZA, C., 2010, "Tool support for collaborative software development based on dependency analysis". In: *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, p. 1–10, Chicago, Illinois, USA.
- COSTA, J. M.; CATALDO, M.; DE SOUZA, C. R., 2011, "The scale and evolution of coordination needs in large-scale distributed projects: implications for the future generation of collaborative tools". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 3151–3160, New York, NY, USA.

- CROSS, R.; ANDREW, P.; CROSS, R., 2004, *The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations*. Boston, Massachusetts, Harvard Business School Press.
- CUBRANIC, D.; BOOTH, K. S., 1999, "Coordinating open-source software development". In: *International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, p. 61–66, Stanford, USA.
- CUGOLA, G.; GHEZZI, C., 1998, "Software processes: A retrospective and a path to the future", *Software Process Improvement and Practice Journal*, v. 4, n. 3, p. 101–123.
- DAMIAN, D.; KWAN, I.; MARCZAK, S., 2010, "Requirements-Driven Collaboration: Leveraging the Invisible Relationships between Requirements and People", In: Mistrík, I., Grundy, J., Hoek, A., et al. [orgs.] (eds), *Collaborative Software Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 57–76.
- DAVID, J. M. N., 2004, *Um Serviço de Percepção para uma Infra-Estrutura de Desenvolvimento de Groupware*. Tese de Doutorado, UFRJ, Rio de Janeiro.
- DAVIS, G. B., 1982, "Strategies for information requirements determination", *Enterprise Analysis - IBM Systems Journal*, v. 21, n. 1, p. 4.
- DEAN, D. L.; DEOKAR, A.; TER BUSH, R., 2006, "Making the Collaboration Engineering Investment Decision". In: *Hawaii International Conference on System Sciences (HICSS)*, p. 1–10, Kauai, Hawaii, USA.
- DEELSTRA, S.; SINNEMA, M.; BOSCH, J., 2005, "Product derivation in software product families: a case study", *Journal of Systems and Software (JSS)*, v. 74, n. 2, p. 173–194.
- DEGENNE, A.; FORSE, M., 1999, *Introducing Social Networks*. 1 ed. Sage Publications.
- DEMARCO, T.; LISTER, T., 1999, *Peopleware: Productive Projects and Teams*. 2 ed. New York, USA, Dorset House.
- DEY, A. K.; ABOWD, G. D.; SALBER, D., 2001, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications", *Human-Computer Interaction*, v. 16, n. 2-4, p. 97–166.

- DIAS, M. DE S., 1998, *COPSE: Um Ambiente de Suporte ao Projeto Cooperativo de Software*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- DOURISH, P.; BELLOTTI, V., 1992, "Awareness and coordination in shared workspaces". In: *Conference on Computer Supported Cooperative Work (CSCW)*, p. 107–114, Toronto, Ontario, Canada.
- DURILLO, J. J.; NEBRO, A. J.; LUNA, F.; *et al.*, 2006, *JMetal: A Java Framework for Developing Multi-objective Optimization Metaheuristics*, Technical Report TR ITI-2006-10, Dept. de Lenguajes y Ciencias de Computacion, Univ. Málaga.
- DYBAA, T.; SJØBERG, D. I. K.; CRUZES, D. S., 2012, "What works for whom, where, when, and why?: on the role of context in empirical software engineering". In: *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, p. 19–28, New York, NY, USA.
- EBERT, C., 2007, "Open Source Drives Innovation", *IEEE Software*, v. 24, n. 3, p. 105–109.
- EGLER, T. T. C., 2007, *Ciberpólis: redes no governo da cidade*. Rio de Janeiro, RJ, Brasil, 7 Letras.
- EHMS, K.; LANGEN, M., 2002, "Holistic development of knowledge management with KMMM", *Siemens AG*
- ELLIS, C. A.; GIBBS, S. J.; REIN, G., 1991, "Groupware: some issues and experiences", *Communications of ACM*, v. 34, n. 1, p. 39–58.
- EVOLTRACK-SOCIALNETWORK, 2013. *EvoTrack-SocialNetwork Homepage*. Disponível em: <http://reuse.cos.ufrj.br/evoltrack/socialnetwork>. Acesso em: 26 ago 2010.
- FEI DAI; TONG LI, 2007, "Tailoring Software Evolution Process". *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD)*, p. 782–787, Nagoya, Japan.
- FEILER, P.; HUMPHREY, W. S., 1992, *Software Process Development and Enactment: Concepts and Definitions*, Technical Report CMU/SEI-92-TR-004, SEI-CMU.

- FELLER, J.; FITZGERALD, B., 2001, *Understanding Open Source Software Development*. Boston, MA, USA, Addison-Wesley.
- FERNANDES, P. C. C., 2008, *UbiFEX: Uma Abordagem para Modelagem de Características de Linha de Produtos de Software Sensíveis ao Contexto*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- FERNANDES, P.; WERNER, C.; MURTA, L. G. P., 2008, "Feature modeling for context-aware software product lines". In: *International Conference on Software Engineering and Knowledge Engineering (SEKE)*, p. 758–763
- FERREIRA, A. B. DE H., 2009, *Novo Dicionário Aurélio da Língua Portuguesa*. 3 ed. Curitiba, Editora Positivo.
- FIORINI, S. T., 2001, *Arquitetura para Reutilização de Processos de Software*. Tese de Doutorado, PUC-RJ, Rio de Janeiro, RJ, Brasil. Disponível em: <http://www-di.inf.puc-rio.br/~julio/tese-soeli.pdf>.
- FISHER, D. M., 2004, "The Business Process Maturity Model. A Practical Approach for Identifying Opportunities for Optimization", *Business Process Trends*
- FITZGERALD, B.; RUSSO, N. L.; O'KANE, T., 2003, "Software development method tailoring at Motorola", *Communications of ACM*, v. 46, n. 4, p. 64–70.
- FREEMAN, L. C., 1978, "Centrality in social networks conceptual clarification", *Social Networks*, v. 1, n. 3, p. 215–239.
- FREITAS, C. M. D. S.; NEDEL, L. P.; GALANTE, R.; *et al.*, 2008, "Extração de Conhecimento e Análise Visual de Redes Sociais". *Seminário Integrado de Hardware e Software (SEMISH) - Congresso da Sociedade Brasileira de Computação (SBC)*, p. 106–120, Belém, PA, Brasil.
- FSF, 2008. The Free Software Definition. Disponível em: <http://www.gnu.org/philosophy/free-sw.html>. Acesso em: 31 maio 2008.
- FUGGETTA, A., 2000, "Software process: a roadmap". In: *Proceedings of the Conference on The Future of Software Engineering*, p. 25–34, Limerick, Ireland.

- FUKS, H.; GEROSA, M. A.; PIMENTEL, M. G., 2003a, "Projeto de Comunicação em Groupware: Desenvolvimento, Interface e Utilização". *Jornada de Atualização em Informática (JAI)*, p. 295–338, Campinas, SP, Brasil.
- FUKS, H.; RAPOSO, A. B.; GEROSA, M. A., 2003b, "Do Modelo de Colaboração 3C à Engenharia de Groupware". *Simpósio Brasileiro de Sistemas Multimídia e Web (WEBMIDIA) – Trilha Trabalho Cooperativo Assistido por Computador (CSCW)*, p. 1–8, Salvador, BA, Brasil.
- FUKS, H.; RAPOSO, A. B.; GEROSA, M. A.; *et al.*, 2011, "Teorias e modelos de colaboração", *Sistemas Colaborativos*, 1 edRio de Janeiro, RJ, Brasil: Elsevier
- GACEK, C.; ARIEF, B., 2004, "The Many Meanings of Open Source", *IEEE Software*, v. 21, n. 1, p. 34–40.
- GAO, Y.; FREEH, V.; MADEY, G., 2003, "Analysis and Modeling of Open Source Software Community". *North American Association for Computational Social and Organization Sciences Conference (NAACSOS)*, p. 1–4, Pittsburgh, PA, USA.
- GARG, A.; CRITCHLOW, M.; PING CHEN; *et al.*, 2003, "An environment for managing evolving product line architectures". *International Conference on Software Maintenance (ICSM)*, p. 358–367, Amsterdam, The Netherlands.
- GARY, K. A.; LINDQUIST, T. E., 1999, "Cooperating process components". In: *International Computer Software and Applications Conference (COMPSAC)*, p. 218–223, Phoenix, AZ, USA.
- GEROSA, M. A.; FUKS, H.; LUCENA, C. J. P., 2003, "Suporte à Percepção em Ambientes de Aprendizagem Colaborativa", *Revista Brasileira de Informática na Educação (RBIE)*, v. 11, n. 2, p. 75–85.
- GEROSA, M.; PIMENTEL, M.; FUKS, H.; *et al.*, 2006, "Development of groupware based on the 3C collaboration model and component technology", *Groupware: Design, Implementation, and Use*, p. 302–309.
- GINSBERG, M.; QUINN, L., 1995, *Process Tailoring and the Software Capability Maturity Model* CMU/SEI-94-TR-024, SEI-CMU. Disponível em:
<http://www.sei.cmu.edu/publications/documents/94.reports/94.tr.024.html>

- GLASS, R. L., 2001, "Agile Versus Traditional: Make Love, Not War!", *Cutter IT Journal*, v. 14, n. 12, p. 12–18.
- GLAZER, H., 2010, "Love and Marriage: CMMI and Agile Need Each Other"
- GLAZER, H.; DALTON, J.; ANDERSON, D.; *et al.*, 2008, *CMMI or Agile: Why Not Embrace Both!*, SEI-CMU. Disponível em: <http://www.sei.cmu.edu/publications/documents/08.reports/08tn003.html>.
- GOLDBERG, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*. 1 ed. Addison-Wesley Professional.
- GOTO, C. S.; ROSA, M. P.; DE SOUZA, C. DE, 2008, "Um Estudo Exploratório sobre os Efeitos da Refatoração na Coordenação das Atividades de Desenvolvimento de Software Livre". *Workshop de Manutenção de Software Moderna (WMSWM)*, p. 1–8, Florianópolis, SC, Brasil.
- GREIF, I., 1988, *Computer-Supported Cooperative Work: A Book of Readings*. Morgan Kaufmann.
- GUTWIN, C.; GREENBERG, S., 1999, *A framework of awareness for small groups in shared-workspace groupware*, Technical Report 99-1, Department of Computer Science, University of Saskatchewan.
- HAEFLIGER, S.; VON KROGH, G.; SPAETH, S., 2007, "Code Reuse in Open Source Software", *Management Science*, v. 54, n. 1, p. 180–193.
- HAIN, S., 2010, "Developing a Situational Maturity Model for Collaboration (SiMMCo) – Measuring Organizational Readiness". *Global Perspectives on Design Science Research (DESRIST) - Doctoral Symposium*, p. 1–6, St.Gallen, Switzerland.
- HAIN, S.; BACK, A., 2009, *State-of-the-Art on Maturity Models for Collaboration*, Technical Report, Universität St. Gallen.
- HAIN, S.; BACK, A., 2011, "Towards a Maturity Model for E-Collaboration - A Design Science Research Approach", *Proceedings of the 2011 44th Hawaii International Conference on System Sciences*, p. 1–10.
- HAINES, R.; VEHRING, N., 2012, "Increasing Team Coordination and Social Motivation Through Awareness Practices: A Case Study". In: *European Conference on Information Systems (ECIS)*, p. 1–12, Barcelona, Spain.

- HANSEN, M. T., 2009, "When Internal Collaboration Is Bad for Your Company", *Harvard Business Review*, v. 84, n. 3, p. 83–88.
- HANSSON, C.; DITTRICH, Y.; GUSTAFSSON, B.; *et al.*, 2006, "How agile are industrial software development practices?", *Journal of Systems and Software (JSS)*, v. 79, n. 9, p. 1295–1311.
- HARDINGHAM, A., 2000, *Trabalho em Equipe*. São Paulo, SP, Brasil, Nobel.
- HARMAN, M., 2007, "The Current State and Future of Search Based Software Engineering". In: *2007 Future of Software Engineering*, p. 342–357
- HARMAN, M., 2010, "Why the virtual nature of software makes it ideal for search based optimization". In: *Proceedings of the 13th international conference on Fundamental Approaches to Software Engineering*, p. 1–12, Berlin, Heidelberg.
- HARMAN, M.; JONES, B. F., 2001, "Search-Based Software Engineering", *Information and Software Technology*, v. 43, p. 833–839.
- HARMAN, M.; MANSOURI, S. A.; ZHANG, Y., 2009, *Search based software engineering: A comprehensive analysis and review of trends techniques and applications*, Technical Report
- HARMSSEN, F.; BRINKKEMPER, S.; OEI, J. L. H., 1994, "Situational method engineering for informational system project approaches". In: *Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle*, p. 169–194
- HEALY, K.; SCHUSSMAN, A., 2003, "The Ecology of Open Source Software". In: *Annual Meeting of the American Sociological Association*, p. 1–20, Atlanta, GA, USA.
- HENDERSON-SELLERS, B., 2002, "Agile or Rigorous OO Methodologies: Getting the Best of Both Worlds", *Cutter IT Journal*, v. 15, n. 1, p. 25–33.
- HENDERSON-SELLERS, B.; RALYTÉ, J., 2010, "Situational method engineering: state-of-the-art review", *Journal of Universal Computer Science (JUICS)*, v. 16, n. 3, p. 424–478.
- DEN HENGST, M.; DEAN, D. L.; KOLFSCHOTEN, G.; *et al.*, 2006, "Assessing the Quality of Collaborative Processes". *Proceedings of the 39th Annual*

Hawaii International Conference on System Sciences (HICSS), p. 1–10, Hyatt Regency, Kauai, Hawaii, USA.

HERBSLEB, J. D.; MOCKUS, A.; FINHOLT, T. A.; *et al.*, 2000, "Distance, dependencies, and delay in a global collaboration". In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, p. 319–328, Philadelphia, Pennsylvania, United States.

HERBSLEB, J. D.; PAULISH, D. J.; BASS, M., 2005, "Global software development at siemens: experience from nine projects". In: *Proceedings of the 27th international conference on Software engineering*, p. 524–533, St. Louis, MO, USA.

HIGHSMITH, J., 2001, "The Great Methodologies Debate: Part 1", *Cutter IT Journal*, v. 14, n. 12, p. 1–44.

HIGHSMITH, J.; COCKBURN, A., 2001, "Agile Software Development: The Business of Innovation", *IEEE Computer*, v. 34, n. 9, p. 120–127.

HODA, R.; NOBLE, J.; MARSHALL, S., 2010, "Balancing acts: walking the Agile tightrope". In: *ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, p. 5–12, New York, NY, USA.

HOLLENBACH, C.; FRAKES, W., 1996, "Software Process Reuse in an Industrial Setting". In: *International Conference on Software Reuse (ICSR)*, p. 22–30

HUMPHREY, W. S., 1989, *Managing the Software Process*. Boston, MA, USA, Addison-Wesley.

IBM, 2009. Rational Unified Process (RUP). Disponível em: <http://www-01.ibm.com/software/awdtools/rup/>. Acesso em: 12 abr 2010.

ISO/IEC, 1995, *ISO/IEC 12207: Information technology - Software life cycle processes*. The International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC.

ISO/IEC, 2007, *ISO/IEC 15939: Software Engineering – Software Measurement Process*. Genebra, Suíça, The International Organization for Standardization and the International Electrotechnical Commission.

- JAKOBSEN, A. B., 2000, "Software processes: live and let die", *IEEE Software*, v. 17, n. 3 (jun.), p. 71–75.
- JAUFMAN, O.; MUNCH, J., 2005, "Acquisition of a Project-Specific Process", *Product Focused Software Process Improvement*, Berlin / Heidelberg: Springer Verlag, p. 328–342.
- JIANG, L.; EBERLEIN, A., 2008, "Towards a framework for understanding the relationships between classical software engineering and agile methodologies". In: *International Workshop on scrutinizing agile practices or shoot-out at the agile corral*, p. 9–14, Leipzig, Germany.
- JIN XU; YONGQIN GAO; CHRISTLEY, S.; *et al.*, 2005, "A Topological Analysis of the Open Source Software Development Community". In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS)*, p. 198–208, Waikoloa, HI, USA.
- KAHKONEN, T.; ABRAHAMSSON, P., 2004, "Achieving CMMI Level 2 with Enhanced Extreme Programming Approach", *Product Focused Software Process Improvement*, Heidelberg: Springer-Verlag, p. 378–392.
- KANG, K.; COHEN, S.; HESS, J.; *et al.*, 1990, *Feature-Oriented Domain Analysis* CMU/SEI-90-TR-21, CMU-SEI. Disponível em: <http://www.sei.cmu.edu/domain-engineering/FODA.html>.
- KARLSTRÖM, D.; RUNESON, P., 2002, "Decision support for extreme programming introduction and practice selection". In: *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, p. 835–841, Ischia, Italy.
- KELLNER, M. I., 1996, "Connecting reusable software process elements and components". *International Software Process Workshop (ISPW)*, p. 8–11, Dijon, France.
- KHOSHAFIAN, S.; BUCKIEWICZ, M., 1995, *Introduction to Groupware, Workflow, and Workgroup Computing*. Wiley.
- KITCHENHAM, B., 2004, *Procedures for Performing Systematic Reviews*, Technical Report TR/SE-0401, Software Engineering Group, Department of Computer Science, Keele University. Disponível em: http://www.idi.ntnu.no/emner/empse/papers/kitchenham_2004.pdf.

- KRAUT, R. E.; STREETER, L. A., 1995, "Coordination in software development", *Commun. ACM*, v. 38, n. 3 (mar.), p. 69–81.
- KUMAR, K.; WELKE, R. J., 1992, "Methodology Engineering: a proposal for situation-specific methodology construction", *Challenges and strategies for research in systems development*, John Wiley & Sons, p. 257–269.
- LAANTI, M.; KETTUNEN, P., 2005, "How to Steer an Embedded Software Project: Tactics for Selecting Agile Software Process Models", *Information and Software Technology*, v. 47, n. 9, p. 587–608.
- LAMPE, C.; STEINFELD, C.; ELLISON, N. B., 2009, "Social network sites and society: current trends and future possibilities", *interactions*, v. 16, n. 1, p. 6–9.
- LEE, K.; KANG, K. C.; LEE, J., 2002, "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering". In: *Proceedings of the 7th International Conference on Software Reuse: Methods, Techniques, and Tools*, p. 62–77
- LEITE, A. M. S., 2011, *Modelo de Contexto para Adaptação de Processos de Software*. Master Thesis, PPGI–UNIRIO, Rio de Janeiro, Brazil (In Portuguese).
- LINDEN, F. J.; SCHMID, K.; ROMMES, E., 2007, *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. 1 ed. Berlin / Heidelberg, Springer.
- LINDVALL, M.; BASILI, V.; BOEHM, B.; *et al.*, 2002, "Empirical Findings in Agile Methods", *Extreme Programming and Agile Methods (XP/Agile Universe)*, Berlin / Heidelberg: Springer-Verlag, p. 81–92.
- LINDVALL, M.; RUS, I., 2000, "Process diversity in software development", *Software, IEEE*, v. 17, n. 4 (ago.), p. 14–18.
- LITTLE, T., 2005, "Context-adaptive agility: managing complexity and uncertainty", *IEEE Software*, v. 22, n. 3, p. 28–35.
- LOPEZ-FERNANDEZ, L.; ROBLES, G.; GONZALEZ-BARAHONA, J. M.; *et al.*, 2004, "Applying Social Network Analysis to the Information in CVS Repositories". *Mining Software Repositories Workshop - International Conference on Software Engineering (ICSE)*, p. 101–105, Edinburgh, Scotland.

- MACHADO, L. F. D. C., 2000, *Modelo para Definição de Processos de Software na Estação TABA*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MADEY, G.; FREEH, V.; TYNAN, R., 2002, "The open source software development phenomenon: An analysis based on social network theory". *Americas Conference on Information Systems (AMCIS)*, p. 1806–1813, Dallas, TX, USA.
- MAGDALENO, A. M., 2006, *Explicitando a Colaboração em Organizações através da Modelagem de Processos de Negócios*. Dissertação de Mestrado, UFRJ/IM/NCE, Rio de Janeiro, RJ, Brasil.
- MAGDALENO, A. M., 2010a, "Balancing Collaboration and Discipline in Software Development Processes". In: *Doctoral Symposium of International Conference on Software Engineering (ICSE)*, p. 331–332, Cape Town, South Africa.
- MAGDALENO, A. M., 2010b, *Apoio à Decisão para o Balanceamento de Colaboração e Disciplina nos Processos de Desenvolvimento de Software*. Exame de Qualificação, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MAGDALENO, A. M., 2010c, "An optimization-based approach to software development process tailoring". In: *PhD Track - International Symposium on Search Based Software Engineering (SSBSE)*, p. 40–43, Benevento, Italy.
- MAGDALENO, A. M.; ARAUJO, R. M.; BORGES, M. R. S., 2007a, "Designing Collaborative Processes". *Workshop on Business Process Modeling, Development, and Support (BPMDS)*, p. 283–290, Trondheim, Norway.
- MAGDALENO, A. M.; ARAUJO, R. M.; BORGES, M. R. S., 2009a, "A Maturity Model to Promote Collaboration in Business Processes", *International Journal of Business Process Integration and Management (IJBPIIM)*, v. 4, n. 2, p. 111–123.
- MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L., 2011a, "A Roadmap to the Collaboration Maturity Model (CollabMM) Evolution". *International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, p. 105–112, Lausanne, Switzerland.
- MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L., 2012a, "COMPOOTIM: An Approach to Software Processes Composition and

Optimization". *Congresso Ibero-Americano em Engenharia de Software (CibSE)*, p. 1–14, Buenos Aires, Argentina.

MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L., 2012b, "An Exploratory Study on Collaboration Understanding in Software Development Social Networks", *International Conference on Collaboration and Technology (CRIWG)*, , chapter 7493, Raesfeld, Germany: Springer, p. 113–120.

MAGDALENO, A. M.; BARROS, M. DE O.; WERNER, C. M. L.; *et al.*, 2010a, "Formulando a Adaptação de Processos de Desenvolvimento de Software como um Problema de Otimização". *I Workshop Brasileiro de Otimização em Engenharia de Software (WOES)*, p. 56–64, Salvador, BA, Brasil.

MAGDALENO, A. M.; CAPPELLI, C.; BAIÃO, F.; *et al.*, 2008a, "A Practical Experience in Designing Business Processes to Improve Collaboration". *Business Process Design (BPD)*, p. 156–168, Brisbane, Australia.

MAGDALENO, A. M.; CAPPELLI, C.; BAIÃO, F. A.; *et al.*, 2008b, "Towards Collaboration Maturity in Business Processes: An Exploratory Study in Oil Production Processes", *Information Systems Management (ISM)*, v. 25, n. 4, p. 302–318.

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2009b, *Revisão Quasi-Sistemática da Literatura: Conciliação de processos de desenvolvimento de software*, Relatório Técnico ES-730, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2010b, *Estudo de Ferramentas de Mineração, Visualização e Análise de Redes Sociais*, Relatório Técnico, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2010c, "Analyzing Collaboration in Software Development Processes through Social Networks". In: *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, p. 435–446, Heraklion, Crete, Greece.

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2011b, *A Criação de um Exemplo para Alimentar a Abordagem COMPOOTIM de Composição e Otimização de Processos de Software*, Relatório Técnico ES-744/11, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.

- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2011c, *Um Roadmap para a Evolução do Modelo de Maturidade em Colaboração (CollabMM)*, Relatório Técnico ES-742/11, PESC-COPPE. Disponível em: <http://www.cos.ufrj.br>.
- MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M., 2012c, "Reconciling Software Development Models: A Quasi-Systematic Review", *Journal of Systems and Software (JSS)*, v. 85, n. 2, p. 351–369.
- MAGDALENO, A.; NUNES, V.; ARAUJO, R.; *et al.*, 2007b, "Flexible Organizational Process Deployment", *Computer Supported Cooperative Work in Design III*, , chapter 4402, Berlin / Heidelberg: Springer, p. 679–688.
- MARCAL, A. S. C.; BEZERRA, C. I. M.; COELHO, C.; *et al.*, 2010, "Uso de Práticas Ágeis para Alcançar o CMMI 5: Uma Abordagem Inovadora". *Simpósio Brasileiro de Qualidade de Software (SBQS)*, p. 343–350, Belém, PA, Brasil.
- MARCAL, A. S. C.; FURTADO, M. E. S., 2010, "Scrummi: Um processo de gestão ágil baseado no Scrum e aderente ao CMMI". *Simpósio Brasileiro de Qualidade de Software (SBQS) - Concurso de Teses e Dissertações em Qualidade de Software*, p. 425–439, Belém, PA, Brasil.
- MARQUES JUNIOR, C. R.; CIRILO, E. J. R.; LUCENA, C. J. P., 2011, "Assisted User-Guidance in Collaborative and Dynamic Software Product Line Configuration". *Congresso Ibero-Americano em Engenharia de Software (CIBSE)*, p. 143–156, Rio de Janeiro, RJ, Brasil.
- MARTÍNEZ-RUIZ, T.; GARCÍA, F.; PIATTINI, M., 2008, "Towards a SPEM v2. 0 Extension to Define Process Lines Variability Mechanisms", *Software Engineering Research, Management and Applications*, p. 115–130.
- MARTÍNEZ-RUIZ, T.; GARCÍA, F.; PIATTINI, M., 2011, "Managing Process Diversity by Applying Rationale Management in Variant Rich Processes", *Product-Focused Software Process Improvement*, , chapter 6759, Springer, p. 128–142.
- MEIER, A.; SPADA, H.; RUMMEL, N., 2007, "A rating scheme for assessing the quality of computer-supported collaboration processes", *International Journal of Computer-Supported Collaborative Learning*, v. 2, n. 1 (fev.), p. 63–86.

- MEIRA, S. R. DE L.; COSTA, R. A.; JUCÁ, P. M.; *et al.*, 2011, "Redes Sociais", *Sistemas Colaborativos*, 1 ed Rio de Janeiro, RJ, Brasil: Elsevier
- MELLO, R. M. DE; TEIXEIRA, E. N.; SCHOTS, M.; *et al.*, 2012, "Checklist-Based Inspection Technique for Feature Models Review". In: *2012 Sixth Brazilian Symposium on Software Components, Architectures and Reuse*, p. 140–149, Los Alamitos, CA, USA.
- MELLOR, S. J.; JACOBSON, I.; HENDERSON-SELLERS, B., 2002, "The Great Methodologies Debate: Part 2", *Cutter IT Journal*, v. 15, n. 1, p. 1–40.
- MILLER, G. G., 2001, "The Characteristics of Agile Software Processes". In: *Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS)*, p. 1–3, Santa Barbara, CA, USA.
- MINICUCCI, A., 2001, *Técnicas de Trabalho de Grupo*. 3a. ed. São Paulo, SP, Brasil, Atlas.
- MISTRİK, I.; GRUNDY, J.; HOEK, A.; *et al.*, 2010, *Collaborative Software Engineering*. Berlin, Heidelberg, Springer.
- MNKANDLA, E., 2008, *A framework for agile methodology practices: a family of methodologies approach*. Doctoral thesis, Faculty of Engineering and the Built Environment, University of The Witwatersrand, Johannesburg. Disponível em: <http://hdl.handle.net/10539/5666>.
- MONCLAR, R. S.; OLIVEIRA, J.; DE FARIA, F. F.; *et al.*, 2011, "Using social networks analysis for collaboration and team formation identification". In: *Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on*, p. 562 –569
- MONTERO, I.; PENA, J.; RUIZ-CORTÉS, A., 2007, "Business Family Engineering: Does it make sense?". In: *I JISBD Taller sobre Procesos de Negocio e Ingenieria del Software (PNIS)*, p. 34–40, Zaragoza, España.
- MOREL, C. M.; SERRUYA, S. J.; PENNA, G. O.; *et al.*, 2009, "Co-authorship Network Analysis: A Powerful Tool for Strategic Planning of Research, Development and Capacity Building Programs on Neglected Diseases", *PLoS Negl Trop Dis*, v. 3, n. 8, p. e501.

- MURTA, L. G. P., 2002, *Charon: Uma Máquina de Processos Extensível Baseado em Agentes Inteligentes*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- NERUR, S.; MAHAPATRA, R.; MANGALARAJ, G., 2005, "Challenges of migrating to agile methodologies", *Communications of ACM*, v. 48, n. 5, p. 72–78.
- NETTO, F. DE C., 2010, *Um Método Automático para Geração de Cronogramas de Tarefas de Correção de Bugs*. Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, RJ, Brasil.
- NEWMAN, M. E. J., 2003, "The structure and function of complex networks", *SIAM Review*, v. 45, n. 2 (mar.), p. 167–256.
- NIELSEN, P.; TJØRNEHØJ, G., 2005, "Mapping Social Networks in Software Process Improvement: An Action Research Study", *Business Agility and Information Technology Diffusion*, Atlanta, Georgia, USA:, p. 73–90.
- NONAKA, I.; TAKEUCHI, H., 1995, *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, USA.
- NOOY, W. DE; MRVAR, A.; BATAGELJ, V., 2005, *Exploratory Social Network Analysis with Pajek*. Cambridge, United Kingdom, Cambridge University Press.
- NORTHROP, L. M., 2002, "SEI's software product line tenets", *IEEE Software*, v. 19, n. 4, p. 32–40.
- NUNAMAKER, J.; ROMANO, N.; BRIGGS, R., 2001, "A Framework for Collaboration and Knowledge Management". In: *Annual Hawaii International Conference on System Sciences (HICSS)*, p. 1060, Maui, HI, USA.
- NUNES, E.; BARRETO, A.; ROCHA, A. R.; *et al.*, 2010a, "Definição de Processos de Aquisição de Software para Reutilização". *Conferência Latino-americana de Informática (CLEI)*, Asunção, Paraguai.
- NUNES, E. D., 2011a, *Definição de Processos de Aquisição de Software para Reutilização*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro.

- NUNES, V. T., 2011b, *Adaptação Dinâmica de Processos: Uma Abordagem Orientada a Contexto*. Exame de Qualificação, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- NUNES, V. T.; WERNER, C. M. L.; SANTORO, F. M., 2011, "Dynamic process adaptation: A context-aware approach". In: *2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, p. 97 –104
- NUNES, V. T.; WERNER, C. M. L.; SANTORO, F. M., 2012, "Mediating process adaptation through a goal-oriented context-aware approach". In: *2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, p. 160 – 167
- NUNES, V. T.; WERNER, C.; SANTORO, F. M., 2010b, "Context-Based Process Line". In: *International Conference on Enterprise Information Systems (ICEIS)*, p. 277–282, Funchal, Madeira, Portugal.
- ODYSSEY, 2013. Odyssey SDE Homepage. Disponível em: <http://reuse.cos.ufrj.br>. Acesso em: 26 ago 2010.
- OEZBEK, C.; PRECHELT, L.; THIEL, F., 2010, "The onion has cancer: some social network analysis visualizations of open source project communication". In: *Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*, p. 5–10, New York, NY, USA.
- OLIVEIRA, E. A., 2012, *Sobre a Colaboração na Comunidade de Sistemas de Informação através dos Simpósios SBSI*. Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, RJ, Brasil.
- OLIVEIRA, F. F., 2009, *Uma Ontologia de Colaboração e suas Aplicações*. Dissertação de Mestrado, Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, Vitória, ES, Brasil.
- OLIVEIRA, M. S. DE, 2011, *PREViA: Uma Abordagem para a Visualização da Evolução de Modelos de Software*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro.

- OLIVEIRA, R. F. DE, 2006, *Formalização e Verificação de Consistência na Representação de Variabilidades*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- OMG, 2008. Software Process Engineering Metamodel. Disponível em: <http://www.omg.org/technology/documents/formal/spem.htm>. Acesso em: 23 ago 2008.
- OMG, 2009. Business Process Management Notation (BPMN) Version 1.2. Disponível em: <http://www.bpmn.org/>. Acesso em: 8 jun 2009.
- OMORONYIA, I.; FERGUSON, J.; ROPER, M.; *et al.*, 2010, "A review of awareness in distributed collaborative software engineering", *Software: Practice and Experience*, v. 40, n. 12, p. 1107–1133.
- ORR, K., 2002, "CMM Versus Agile Development: Religious Wars and Software Development", *Cutter Consortium*, v. 3, n. 7, p. 1–34.
- OSTERWEIL, L., 1987, "Software processes are software too". In: *International Conference on Software Engineering (ICSE)*, p. 2–13, Monterey, CA, USA.
- PAI, M.; MCCULLOCH, M.; GORMAN, J. D.; *et al.*, 2004, "Systematic reviews and meta-analyses: an illustrated, step-by-step guide", *The National Medical Journal of India*, v. 17, n. 2, p. 86–95.
- PALMER, S. R.; FELSING, J. M., 2002, *A Practical Guide to Feature-Driven Development*. USA, Prentice Hall.
- PAPADIMITRIOU, C. H.; STEIGLITZ, K., 1998, *Combinatorial Optimization: Algorithms and Complexity*. Unabridged ed. New York, USA, Dover Publications.
- PARK, S.; NA, H.; PARK, S.; *et al.*, 2006, "A semi-automated filtering technique for software process tailoring using neural network", *Expert Systems with Applications*, v. 30, n. 2 (fev.), p. 179–189.
- PATEL, C.; LYCETT, M.; MACREDIE, R.; *et al.*, 2006, "Perceptions of Agility and Collaboration in Software Development Practice". In: *Hawaii International Conference on System Sciences (HICSS)*, p. 1–7, Kauai, Hawaii, USA.

- PAULK, M. C., 2001, "Extreme programming from a CMM perspective", *Software, IEEE*, v. 18, n. 6, p. 19–26.
- PAULK, M. C., 2009, *A History of the Capability Maturity Model for Software*, Technical Report, American Society for Quality (ASQ). Disponível em: www.asq.org.
- PAULK, M. C.; WEBER, C. V.; CURTIS, B.; *et al.*, 1994, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Boston, MA, USA, Addison-Wesley.
- PEDREIRA, O.; PIATTINI, M.; LUACES, M. R.; *et al.*, 2007, "A systematic review of software process tailoring", *SIGSOFT Software Engineering Notes*, v. 32, n. 3, p. 1–6.
- PEREIRA, C. S.; SOARES, A. L., 2007, "Improving the quality of collaboration requirements for information management through social networks analysis", *International Journal of Information Management*, v. 27, n. 2 (abr.), p. 86–103.
- PEREIRA, G. V., 2012, *Abordagem Multicritérios para Adaptação de Processos de Software Baseada Em Situational Method Engineering*. Dissertação de Mestrado, Universidade Federal de Santa Maria (UFSM), Santa Maria, RS, Brasil.
- PEREIRA, G. V.; SEVERO, F.; FONTOURA, L., 2012, "A Risk Management Approach Based on Situational Method Engineering". In: *Congresso Ibero-Americano em Engenharia de Software (CibSE) - Short papers*, p. 1–6, Buenos Aires, Argentina.
- PIMENTEL, M.; FUKS, H., 2011, *Sistemas Colaborativos*. 1 ed. Rio de Janeiro, RJ, Brasil, Elsevier.
- PINZGER, M.; GALL, H. C., 2010, "Dynamic Analysis of Communication and Collaboration in OSS Projects", In: Mistrík, I., Grundy, J., Hoek, A., *et al.* [orgs.] (eds), *Collaborative Software Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, p. 265–284.
- PÖPPELBUß, J.; RÖGLINGER, M., 2011, "What Makes a Useful Maturity Model? A Framework of General Design Principles for Maturity Models and its Demonstration in Business Process Management". *European Conference in Information Systems (ECIS)*, Helsinki, Finland.

- PRIKLADNICKI, R.; MAGALHAES, A. L. C. C., 2010, "Implantação de Modelos de Maturidade com Metodologias Ágeis: Um Relato de Experiências". *VI Workshop Anual do MPS (WAMPS)*, p. 88–98, Campinas, SP, Brasil.
- PUNTAR, S.; IENDRIKE, H. DOS S.; MAGDALENO, A. M.; *et al.*, 2009, *Estudo Conceitual sobre BPMS*, Relatório Técnico 0007/2009, RelaTe-DIA, UNIRIO. Disponível em: <http://seer.unirio.br/index.php/monografiasppgi>.
- QUMER, A.; HENDERSON-SELLERS, B., 2008, "An evaluation of the degree of agility in six agile methods and its applicability for method engineering", *Information and Software Technology*, v. 50, n. 4 (mar.), p. 280–295.
- RAMAN, S., 2000, "It is software process, stupid: next millennium software quality key", *Aerospace and Electronic Systems Magazine, IEEE*, v. 15, n. 6, p. 33–37.
- RAYMOND, E. S., 2001, *The Cathedral & the Bazaar*. Revised & Expanded ed. O'Reilly Media.
- REEVES, C. R., 1993, *Modern Heuristic Techniques for Combinatorial Problems*. Halsted Press.
- REIFER, D. J., 2002, "How Good are Agile Methods?", *IEEE Software*, v. 19, n. 4, p. 16–18.
- REIFER, D. J., 2003, "XP and the CMM", *IEEE Software*, v. 20, n. 3, p. 14–15.
- REIS, C. R., 2003, *Caracterização de um Processo de Software para Projetos de Software Livre*. Dissertação de Mestrado, USP, São Paulo, SP, Brasil. Disponível em: <http://www.async.com.br/~kiko/dissert.pdf>.
- REIS, R. Q., 2002, *APSEE-Reuse: um meta-modelo para apoiar a reutilização de processos de software*. Tese de Doutorado, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brasil. Disponível em: <http://hdl.handle.net/10183/1622>.
- RIBÓ, J. M.; FRANCH, X., 2002, *Supporting Process Reuse in PROMENADE*, Research Report LSI-02-14-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya.

- ROBERTS, F. S., 2009, *Measurement Theory: Volume 7: With Applications to Decisionmaking, Utility, and the Social Sciences (Encyclopedia of Mathematics and its Applications)*. Reissue ed. Cambridge University Press.
- ROCHA, A. R. C. DA; MALDONADO, J. C.; WEBER, K. C., 2001, *Qualidade de Software: Teoria e Prática*. São Paulo, SP, Brasil, Prentice Hall.
- ROMBACH, D., 2006, "Integrated Software Process and Product Lines", *Unifying the Software Process Spectrum*, Heidelberg: Springer-Verlag, p. 83–90.
- RONG, G.; SHAO, D.; ZHANG, H., 2010, "SCRUM-PSP: Embracing Process Agility and Discipline". In: *Asia Pacific Software Engineering Conference*, p. 316–325, Washington, DC, USA.
- ROSEMANN, M.; DE BRUIN, T.; HUEFFNER, T., 2004, "A model for business process management maturity". In: *Australasian Conference on Information Systems (ACIS)*, p. 1–6, Hobart, Australia.
- ROSEMANN, M.; RECKER, J. C., 2006, "Context-aware Process Design: Exploring the Extrinsic Drivers for Process Flexibility".
- SAIDANI, O.; NURCAN, S., 2007, "Towards context aware business process modelling". In: *8th Workshop on Business Process Modeling, Development, and Support (BPMDS)*, p. 265–273, Trondheim, Norway.
- SALGADO, A.; MELCOP, T.; ACCHAR, J.; *et al.*, 2010, "Aplicação de um Processo Ágil para Implantação de Processos de Software baseado em Scrum na Chemtech". *Simpósio Brasileiro de Qualidade de Software (SBQS)*, p. 351–358, Belém, PA, Brasil.
- SANTANA, C.; GUSMÃO, C.; SOARES, L.; *et al.*, 2009, "Agile Software Development and CMMI: What We Do Not Know about Dancing with Elephants", *Agile Processes in Software Engineering and Extreme Programming*, , chapter 31, Heidelberg: Springer, p. 124–129.
- SANTANEN, E.; KOLFSCHOTEN, G.; GOLLA, K., 2006, "The Collaboration Engineering Maturity Model". *39th Annual Hawaii International Conference on System Sciences (HICSS)*, p. 1–10

- SANTORO, F.; BRÉZILLON, P.; ARAUJO, R., 2007, "Context Dynamics in Software Engineering Process", *Computer Supported Cooperative Work in Design (CSCWD)*, Melbourne, Australia, p. 377–388.
- SANTORO, F. M., 2001, *Um Modelo de Cooperação para Aprendizagem baseada em Projetos*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- SANTOS, T. A. L., 2010, *Identificação de Características de Colaboração em Processos de Desenvolvimento de Software através de Redes Sociais*. Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, RJ, Brasil.
- SANTOS, T. A. L.; ARAUJO, R. M. DE; MAGDALENO, A. M., 2009, "Padrões para Percepção da Colaboração em Redes Sociais de Desenvolvimento de Software". *Workshop de Desenvolvimento Distribuído de Software (WDDS)*, p. 1–10, Fortaleza, CE, Brasil.
- SANTOS, T. A. L.; ARAUJO, R. M.; MAGDALENO, A. M., 2010, "Identifying Collaboration Patterns in Software Development Social Networks", *Infocomp - Journal of Computer Science - Special Issue*, p. 51–60.
- SANTOS, T. A. L.; ARAUJO, R. M.; MAGDALENO, A. M., 2011, "Bringing Out Collaboration in Software Development Social Networks". In: *International Conference on Product Focused Software Development and Process Improvement (PROFES) - Short Papers*, p. 18–21, Torre Canne, Italy.
- SANTOS, V. A., 2009, *Aprendizado Organizacional e Melhoria Contínua de Processos de Software através de Reuso de Processos de Software*. Dissertação de Mestrado, Universidade Estadual do Ceará (UEC), Fortaleza, CE, Brasil.
- SARMA, A.; MACCHERONE, L.; WAGSTROM, P.; *et al.*, 2009, "Tesseract: Interactive visual exploration of socio-technical relationships in software development". In: *Software Engineering, International Conference on*, p. 23–33, Los Alamitos, CA, USA.
- SARMENTO, A., 2002, *Impacto dos sistemas colaborativos nas organizações: estudo de casos de adoção e utilização de sistemas workflow*. Tese de Doutorado, Departamento de Sistemas de Informação - Universidade do Minho, Minho, Portugal. Disponível em: <http://repositorium.sdum.uminho.pt/handle/1822/285>.

- SCACCHI, W., 2007, "Free/open source software development: recent research results and emerging opportunities". *Meeting on European software engineering conference*, p. 459–468, Dubrovnik, Croatia.
- SCACCHI, W., 2010, "Collaboration Practices and Affordances in Free/Open Source Software Development", In: Mistrík, I., Grundy, J., Hoek, A., *et al.* [orgs.] (eds), *Collaborative Software Engineering*, Springer Berlin Heidelberg, p. 307–327.
- SCHEER, A.-W., 2000, *ARIS - Business Process Modeling*. 3rd ed. Springer.
- SCHMIDT, K., 2002, "The Problem with "Awareness": Introductory Remarks on "Awareness in CSCW"", *Computer Supported Cooperative Work (CSCW)*, v. 11, n. 3 (nov.), p. 285–298.
- SCHNIEDERS, A.; PUHLMANN, F., 2006, "Variability mechanisms in e-business process families". *International Conference on Business Information Systems (BIS)*, p. 583–601, Klagenfurt, Austria.
- SCHOLTES, P. R.; JOINER, B. L.; STREIBEL, B. J., 2003, *The Team Handbook*. 3rd ed. USA, Joiner/Oriel Inc.
- SCHWABER, K., 2004, *Agile Project Management with Scrum*. Washington, DC, USA, Microsoft Press.
- SCHWIND, M.; WEGMANN, C., 2008, "SVNNAT: Measuring Collaboration in Software Development Networks". In: *IEEE Joint Conference on E-Commerce Technology (CEC) and Enterprise Computing, E-Commerce and E-Services (EEE)*, p. 97–104, Washington, DC, USA.
- SHIRKY, C., 2003. A Group Is Its Own Worst Enemy. Disponível em: http://shirky.com/writings/group_enemy.html. Acesso em: 26 abr 2008.
- SHULL, F.; CARVER, J.; TRAVASSOS, G. H., 2001, "An empirical methodology for introducing software processes", *SIGSOFT Softw. Eng. Notes*, v. 26, n. 5, p. 288–296.
- SIEBEL, N. T.; COOK, S.; SATPATHY, M.; *et al.*, 2003, "Latitudinal and longitudinal process diversity", *Journal of Software Maintenance: Research and Practice*, v. 15, n. 1 (jan.), p. 9–25.

- SILVA, M. A., 2010, *IAVEMS: Infraestrutura de Apoio à Visualização da Evolução de Métricas de Software*. Projeto Final, UFRJ/IM, Rio de Janeiro, RJ, Brasil.
- SLOOTEN, K. VAN; BRINKKEMPER, S., 1993, "A Method Engineering Approach to Information Systems Development". In: *IFIP WG8.1 Working Conference on Information System Development Process*, p. 167–186, Como, Italy.
- SOFTEX, 2011, *Melhoria de Processo do Software Brasileiro – Guia Geral*, Modelo de Qualidade Disponível em: <http://www.softex.br>.
- SOFTEX, 2012, *Melhoria de Processo do Software Brasileiro – Guia Geral de Software*, Modelo de Qualidade Disponível em: <http://www.softex.br>.
- VAN SOLINGEN, R., 2004, "Measuring the ROI of software process improvement", *IEEE Software*, v. 21, n. 3, p. 32–38.
- SOMMERVILLE, I., 2004, *Software Engineering*. 7 ed. Addison Wesley.
- SOUSA JUNIOR, S. F.; BALIEIRO, M. A.; DOS R. COSTA, J. M.; *et al.*, 2009, "Multiple Social Networks Analysis of FLOSS Projects using Sargas". In: *42nd Hawaii International Conference on System Sciences (HICSS)*, p. 1–10, Waikoloa, Big Island, Hawaii, USA.
- SOUZA, C. DE; FROEHLICH, J.; DOURISH, P., 2005, "Seeking the source: software source code as a social and technical artifact". In: *International ACM SIGGROUP Conference on Supporting Group Work*, p. 197–206, Sanibel Island, FL, USA.
- SOUZA, C. R. B.; COSTA, J. M. R.; CATALDO, M., 2012, "Analyzing the scalability of coordination requirements of a distributed software project", *Journal of the Brazilian Computer Society*, v. 18, n. 3 (abr.), p. 201–211.
- SOUZA, C. R. B. DE; MARCZAK, S.; PRIKLADNICKI, R., 2011, "Desenvolvimento colaborativo de software", *Sistemas Colaborativos*, Rio de Janeiro, RJ, Brasil: Elsevier
- SOUZA, C. R. B.; REDMILES, D. F., 2011, "The Awareness Network, To Whom Should I Display My Actions? And, Whose Actions Should I Monitor?", *IEEE Transactions on Software Engineering*, v. 37, n. 3 (jun.), p. 325 – 340.

- SOUZA, C. R. DE; QUIRK, S.; TRAINER, E.; *et al.*, 2007, "Supporting collaborative software development through the visualization of socio-technical dependencies". In: *ACM SIGGROUP Conference on Supporting Group Work*, p. 147–156, Sanibel Island, FL, United States.
- STAPLETON, J.; CONSTABLE, P., 1997, *DSDM Dynamic Systems Development Method: The Method in Practice*. Boston, MA, USA, Addison Wesley.
- STEINMACHER, I.; CHAVES, A. P.; GEROSA, M. A., 2012, "Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature", *Journal of Computer Supported Cooperative Work (JCSCW)*, p. 1–40.
- SUTTON, S. M.; OSTERWEIL, L. J., 1996, "Product families and process families". *International Software Process Workshop (ISPW)*, p. 109–111, Dijon, France.
- SWENSON, K. D.; PALMER, N.; KEMSLEY, S.; *et al.*, 2011, *Social BPM*. Future Strategies Inc.
- SZWARCFITER, J., 1986, *Grafos e Algoritmos Computacionais*. Rio de Janeiro, RJ, Brasil, Campus.
- TATA, 2010, *5iKM3 Knowledge Management Maturity Model: Business Intelligence and Performance Management*, Brochure
- TAURION, C., 2004, *Software Livre: Potencialidades e Modelos de Negócio*. 2 ed. Rio de Janeiro, RJ, Brasil, Brasport.
- TEIXEIRA, E. N., 2011, *OdysseyProcess-FEX: Uma Abordagem para Modelagem de Variabilidades de Linha de Processos de Software*. Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- TELLERIA, K. M.; LITTLE, D.; MACBRYDE, J., 2002, "Managing processes through teamwork", *Business Process Management Journal*, v. 8, n. 4, p. 338–350.
- TERNITE, T., 2009, "Process Lines: A Product Line Approach Designed for Process Model Development". *35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, p. 173–180, Patras, Greece.

- THEUNISSEN, M.; KOURIE, D.; BOAKE, A., 2008, "Corporate-, Agile- and Open Source Software Development: A Witch's Brew or An Elixir of Life?", *Balancing Agility and Formalism in Software Engineering: Second IFIP TC 2 Central and East European Conference on Software Engineering Techniques (CEE-SET)*, Springer-Verlag, p. 84–95.
- THOMSON, A. M.; PERRY, J. L.; MILLER, T. K., 2009, "Conceptualizing and Measuring Collaboration", *Journal of Public Administration Research and Theory*, v. 19, n. 1, p. 23 –56.
- TRAINER, E.; QUIRK, S.; SOUZA, C. DE; *et al.*, 2005, "Bridging the gap between technical and social dependencies with Ariadne". In: *International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, p. 26–30, San Diego, USA.
- TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G., 2002, *Introdução à Engenharia de Software Experimental*, Relatório Técnico ES-590/02, PESC-COPPE.
- TRAVASSOS, G. H.; DOS SANTOS, P. S. M.; NETO, P. G. M.; *et al.*, 2008, "An Environment to Support Large Scale Experimentation in Software Engineering". In: *International Conference on the Engineering of Complex Computer Systems (ICECCS)*, p. 193–202, Belfast, United Kingdom.
- TURK, D.; FRANCE, R.; RUMPE, B., 2002, "Limitations of agile software processes". In: *International Conference on Extreme Programming and Flexible Processes in Software Engineering*, p. 43–46, Alghero, Italy.
- TURNER, R.; JAIN, A., 2002, "Agile Meets CMMI: Culture Clash or Common Cause?", *Extreme Programming and Agile Methods - XP/Agile Universe 2002*, 1 ed, chapter 2418, Heidelberg: Springer-Verlag, p. 153–165.
- VAHIA, C. M.; MAGDALENO, A. M.; WERNER, C. M. L., 2011, "EvoITrack-SocialNetwork: Uma ferramenta de apoio à visualização de redes sociais". In: *Congresso Brasileiro de Software: Teoria e Prática (CBSOFT) – Sessão de Ferramentas*, p. 7–13, São Paulo, SP, Brasil.
- VALETTO, G.; HELANDER, M.; EHRLICH, K.; *et al.*, 2007, "Using Software Repositories to Investigate Socio-technical Congruence in Development Projects". In: *ICSE International Workshop on Mining Software Repositories (MSR)*, p. 25–28, Minneapolis, MN, USA.

- VIEGAS, F.; DONATH, J., 2004, "Social network visualization: Can we go beyond the graph", *Workshop on Social Networks, CSCW*, v. 4, p. 6–10.
- VINEKAR, V.; SLINKMAN, C. W.; NERUR, S., 2006, "Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View", *Information Systems Management (ISM)*, v. 23, n. 3, p. 31–42.
- VIVACQUA, A. S.; GARCIA, A. C. B., 2011, "Ontologia de colaboração", *Sistemas Colaborativos*, Rio de Janeiro, RJ, Brasil: Elsevier
- VREEDE, G.-J.; BRIGGS, R. O., 2005, "Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks". In: *Hawaii International Conference on System Sciences (HICSS)*, p. 1–10, Hilton Waikoloa Village, Hawaii, USA.
- WARSTA, J.; ABRAHAMSSON, P., 2003, "Is Open Source Software Development Essentially an Agile Method?". In: *Proceedings of the Workshop on Open Source Software Development*, p. 143–147, Portland, OR, USA.
- WASHIZAKI, H., 2006, "Building Software Process Line Architectures from Bottom Up", *Product-Focused Software Process Improvement (PROFES)*, , chapter 4034, Amsterdam, The Netherlands: LNCS, p. 415–421.
- WASSERMAN, S.; FAUST, K., 1994, *Social Network Analysis: Methods and Applications*. 1 ed. Cambridge, United Kingdom, Cambridge University Press.
- WEERD, G. C. VAN DE, 2009, *Advancing in software product management: An incremental method engineering approach*. Tese de Doutorado, Utrecht University, Netherlands. Disponível em: <http://igitur-archive.library.uu.nl/dissertations/2009-0826-200144/UUindex.html>.
- WERNER, C. M. L.; ARAUJO, R. M. DE; SANTORO, F. M.; *et al.*, 2011a, *CDSOFT: Balanceando Colaboração e Disciplina em Processos de Desenvolvimento de Software*. Projeto de Pesquisa, Conselho Nacional de Desenvolvimento Científico e Tecnológico, Rio de Janeiro, RJ, Brasil.
- WERNER, C. M. L.; MURTA, L. G. P.; SCHOTS, M.; *et al.*, 2011b, "EvoTrack: A Plug-in-Based Infrastructure for Visualizing Software Evolution". *Workshop Brasileiro de Visualização de Software (WBVS)*, p. 1–8, São Paulo, SP, Brasil.

- WHITEHEAD, J.; MISTRÍK, I.; GRUNDY, J.; *et al.*, 2010, "Collaborative Software Engineering: Concepts and Techniques", *Collaborative Software Engineering*, v. 1, p. 1–30.
- WOHLIN, C.; RUNESON, P.; HÖST, M., 1999, *Experimentation in Software Engineering: An Introduction*. 1st ed. Springer.
- WWF, 2003, *Redes: uma introdução às dinâmicas da conectividade e da auto-organização*, Relatório Técnico, WWF - Brasil. Disponível em: <http://www.wwf.org>.
- XU, P.; RAMESH, B., 2008, "Using Process Tailoring to Manage Software Development Challenges", *IT Professional*, v. 10, n. 4, p. 39–45.
- YAMAUCHI, Y.; YOKOZAWA, M.; SHINOHARA, T.; *et al.*, 2000, "Collaboration with Lean Media: how open-source software succeeds". In: *ACM Conference on Computer Supported Cooperative Work (CSCW)*, p. 329–338, Philadelphia, PA, USA.
- YILDRIM, N., 2006, "Interactions of Software Technology with Technologies and Social Issues: Interfaces and Requirements for Interdisciplinary Collaboration". *Technology Management for the Global Future (PICMET)*, p. 356–367, Istanbul, Turkey.

Apêndices

Apêndice I – Termo de Consentimento Livre Esclarecido do Estudo Exploratório da EvoTrack-SocialNetwork

O termo de consentimento livre esclarecido deve ser entregue antes do início do estudo, de forma a obter o consentimento do participante. Para dar respaldo a tal consentimento, o formulário provê algumas informações acerca do estudo, tais como objetivo e formato.

Avaliação da Percepção da Colaboração em Redes Sociais

Termo de Consentimento Livre Esclarecido

OBJETIVO DO ESTUDO

Este estudo visa avaliar atividades de visualização e análise de redes sociais em relação à percepção da colaboração durante o acompanhamento de projetos de desenvolvimento de software.

IDADE

Eu declaro ter mais de 18 anos de idade e concordo em participar de um estudo conduzido por Andréa Magalhães Magdaleno da COPPE/UFRJ.

PROCEDIMENTO

Neste estudo, os participantes deverão realizar algumas tarefas com base nos diagramas de redes sociais apresentados. Todos os documentos utilizados neste estudo serão apresentados ao participante e deverão ser preenchidos pelo próprio.

CONFIDENCIALIDADE

Eu estou ciente de que meu nome não será divulgado em hipótese alguma. Também estou ciente de que os dados obtidos por meio deste estudo serão mantidos sob confidencialidade, e os resultados serão posteriormente apresentados de forma agregada, de modo que um participante não seja associado a um dado específico.

Da mesma forma, me comprometo a não comunicar os meus resultados enquanto o estudo não for concluído, bem como manter sigilo das técnicas e documentos apresentados e que fazem parte do experimento.

BENEFÍCIOS E LIBERDADE DE DESISTÊNCIA

Eu entendo que, uma vez o experimento tenha terminado, os trabalhos que desenvolvi, serão estudados visando entender a eficiência dos procedimentos e as técnicas que me foram ensinadas.

Os benefícios que receberei deste estudo são limitados ao aprendizado do material que é distribuído e ensinado. Também entendo que sou livre para realizar perguntas a qualquer momento, solicitar que qualquer informação relacionada a minha pessoa não seja incluída no estudo ou comunicar minha desistência de participação. Por fim, declaro que participo de livre e espontânea vontade com o único intuito de contribuir para o avanço e desenvolvimento de técnicas e processos para a Engenharia de Software.

PESQUISADOR RESPONSÁVEL

Andréa Magalhães Magdaleno
Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

PROFESSORES RESPONSÁVEIS

Profa. Cláudia Maria Lima Werner
Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

Profa. Renata Mendes de Araujo
Programa de Pós-Graduação em Informática -PPGI/UNIRIO

Nome (em letra de forma): _____

Assinatura: _____

Data: _____

Apêndice II – Formulário de Caracterização do Participante do Estudo Exploratório da EvolTrack-SocialNetwork

O formulário de caracterização do participante deve ser entregue antes do início do estudo, com o objetivo de caracterizar o perfil de cada participante para auxiliar na análise dos dados obtidos por meio do estudo.

Avaliação da Percepção da Colaboração em Redes Sociais	
Formulário de Caracterização do Participante	
Código do Participante:	

Este formulário contém algumas perguntas sobre sua experiência acadêmica e profissional.

1. Formação acadêmica

- Pós-Doutorado
- Doutorado concluído
- Doutorado em andamento
- Mestrado concluído
- Mestrado em andamento
- Graduação concluída
- Graduação em andamento

Ano de ingresso: _____

Ano de conclusão/Ano previsto de conclusão:

2. Formação geral

a) Grau de Experiência

Por favor, indique o seu grau de experiência nas áreas de conhecimento a seguir, com base na escala abaixo:

Área de Conhecimento	Grau de Experiência				
Engenharia de Software	0	1	2	3	4
Gerência de Projetos de Software	0	1	2	3	4
Colaboração	0	1	2	3	4

0 = nenhum (*nunca participou de atividades deste tipo*)

1 = estudei em aula ou em livro (*possui conhecimento teórico apenas*)

2 = pratiquei em projetos em sala de aula (*possui conhecimento teórico aplicado apenas no contexto acadêmico*)

3 = usei em projetos pessoais (*possui conhecimento teórico somado de experiências práticas individuais*)

4 = usei em projetos na indústria (*possui conhecimento teórico somado de experiências práticas reais*)

b) Tempo de Experiência

Por favor, detalhe sua resposta. Inclua o número de meses de experiência para cada uma das áreas de conhecimento.

Área de Conhecimento	Tempo de Experiência (meses)
Engenharia de Software	
Gerência de Projetos de Software	
Colaboração	

3. Experiência no domínio exemplo

Esta seção será utilizada para compreender quão familiar você está com o domínio que será utilizado para as atividades durante o experimento. Por favor, indique o grau de experiência no domínio seguindo a escala abaixo:

Domínio	Grau de Experiência		
Software Livre	0	1	2

0 = Eu não tenho familiaridade com este domínio.

1 = Eu tenho alguma familiaridade com este domínio, mas não sou um especialista.

2 = Eu sou muito familiar com este domínio.

Comentários:

4. Experiência em ambientes similares

Esta seção será utilizada para compreender quão familiar você está com outras ferramentas de visualização e análise de redes sociais. Por favor, indique o grau de experiência com este tipo de ferramenta seguindo a escala abaixo:

Ferramenta	Grau de Experiência		
Ferramentas de visualização e análise de redes sociais	0	1	2

0 = Eu não tenho familiaridade com este tipo de ferramenta.

1 = Eu utilizo este tipo de ferramenta algumas vezes.

2 = Eu sou muito familiar com este tipo de ferramenta.

Comentários:

Desde já, agradecemos a sua colaboração.

Andréa Magalhães Magdaleno
Cláudia Maria Lima Werner
Renata Mendes de Araujo

Apêndice III – Formulário para Realização do Estudo Exploratório da EvoTrack-SocialNetwork

O formulário para realização do estudo tem por objetivo apresentar as tarefas que devem ser realizadas pelo participante, bem como coletar as suas respostas.

Avaliação da Percepção da Colaboração em Redes Sociais Formulário para Realização do Estudo	
Código do Participante:	

CONTEXTUALIZAÇÃO

Suponha uma situação onde você é o gerente de um projeto de desenvolvimento de software cujo objetivo é criar um produto inovador. Tal característica do produto faz com que os requisitos sejam voláteis e mudem com frequência, tornando qualquer documentação rapidamente obsoleta. Este projeto envolve uma equipe de desenvolvimento geograficamente distribuída, com uma intensa necessidade de se comunicar, coordenar o trabalho sendo feito e compartilhar informações.

Como gerente, a sua tarefa principal é acompanhar o andamento deste projeto, monitorar o desenvolvimento do produto e antecipar possíveis problemas. De acordo com as características do projeto, você já deduziu que a colaboração é um fator importante no projeto e está preocupado em saber como ela está de fato acontecendo no seu projeto.

INSTRUÇÕES

Para a execução desta atividade, siga as instruções abaixo.

- Verifique se o documento com o conjunto de tarefas foi entregue;
- Resolva as tarefas do formulário **na ordem em que elas são apresentadas**;
- Registre o **horário de início** e o **horário de término** de cada atividade sempre que solicitado. Se for gasto algum tempo no entendimento do modelo antes das atividades, este tempo não deve ser contabilizado.

TAREFAS

Responda às seguintes questões utilizando a ferramenta EvoTrack-SocialNetwork e consultando os dados do projeto **Torrent Episode Downloader**:

Etapa 1 – Questões 1 e 2	
Horário de Início:	
Horário de Término:	

A1) Quais atores compõem a **versão 1014** da **rede social**? Escreva os nomes de cada um deles.

A2) Qual é a densidade da **versão 1036** da **rede sócio-técnica**?

Etapa 2 – Questões 3, 4, 5 e 6	
Horário de Início:	
Horário de Término:	

A3) Qual é a distância (em número de arestas) entre a classe TedMainDialog e a classe TedMainToolBarButton na **versão 1036** da **rede técnica**?

A4) Quem tem a maior centralidade de grau na **versão 1036** da **rede sócio-técnica**?

A5) O que mudou na **rede social** da versão 900 para a versão 1014?

A6) Com quem mhstead pode estar em conflito na **versão 900**?

Etapa 3 – Questões 7, 8, 9 e 10	
Horário de Início:	
Horário de Término:	

Usando as versões e os tipos de rede que julgar mais apropriados:

A7) Qual artefato tem indícios de que precisa ser refatorado? Por que?

A8) Para quem você passaria a tarefa de trabalhar no artefato GeneralPanel? Por que?

A9) Com quem ted_jofo precisa se comunicar para coordenar o trabalho?

A10) De acordo com as características abaixo, como você classificaria a coordenação do projeto Torrent Episode Downloader? Por que?

() Coordenação absoluta - Em projetos com coordenação absoluta, é possível encontrar poucos líderes, os quais centralizam as atividades e permeiam as relações com os demais membros do grupo.

() Coordenação múltipla - Em projetos com coordenação múltipla, já existem características de descentralização, multiplicando assim os nós centrais através da rede. Os líderes tendem a dividir cada vez mais as tarefas entre o grupo neste tipo de rede.

() Coordenação distribuída - Em projetos com coordenação distribuída, a principal característica é a tendência à ausência de líderes, pois as atividades do projeto se encontram igualmente distribuídas.

	Coordenação Absoluta	Coordenação Múltipla	Coordenação Distribuída
Densidade da rede	Baixa (0 a 0,30)	Baixo-média a médio-alta (0,31 a 0,70)	Alta (0,71 a 1)
Centralidade de grau da rede	Alta (0,70 a 1)	Médio-alta a baixo-média (0,31 a 0,70)	Baixa (0 a 0,30)
Centralidade de intermediação da rede	Alta a médio-alta (0,51 a 1)	Baixo a baixo-média (0 a 0,50)	Baixa (0 a 0,30)

Obrigada pela sua colaboração.

Andréa Magalhães Magdaleno
Cláudia Maria Lima Werner
Renata Mendes de Araujo

Apêndice IV – Questionário de Avaliação do Estudo Exploratório da EvolTrack-SocialNetwork

O questionário de avaliação do estudo deve ser entregue na última parte do estudo, e tem como objetivo coletar as percepções e considerações sobre a execução das atividades com e sem a aplicação da ferramenta EvolTrack-SocialNetwork.

Avaliação da Percepção da Colaboração em Redes Sociais	
Questionário de Avaliação do Estudo	
Código do Participante:	

Prezado(a) participante,

Esta é a última parte de um estudo de utilização da ferramenta EvolTrack-SocialNetwork. O objetivo deste questionário é obter informações adicionais para a avaliação da ferramenta aplicada no estudo, a partir das respostas às questões listadas a seguir:

1) Você conseguiu efetivamente realizar todas as tarefas propostas?

- () Sim
 () Parcialmente
 () Não

Comentários:

2) Você ficou satisfeito com o resultado final das tarefas?

- () Sim
 () Parcialmente
 () Não

Comentários:

3) No seu ponto de vista, é possível perceber como a colaboração acontece em um projeto de desenvolvimento de software usando as informações apresentadas?

- Sim
- Parcialmente
- Não

Comentários:

4) Em que tipo de projeto ou equipe você considera que a percepção da colaboração pode agregar maior valor?

Comentários:

5) Qual o grau de dificuldade na realização das tarefas?

- A execução das tarefas é muito difícil
- A execução das tarefas é difícil
- A execução das tarefas é fácil
- A execução das tarefas é muito fácil

Comentários:

6) Qual a maior dificuldade encontrada na realização das tarefas?

Comentários:

7) Como a ferramenta **EvoITrack-SocialNetwork** contribuiu com a realização das tarefas?

- Facilitou bastante
- Facilitou um pouco
- Não teve muito impacto (indiferente / não facilitou em nada)
- Dificultou um pouco
- Dificultou bastante

8) Quais as funcionalidades da ferramenta EvoITrack-SocialNetwork que foram mais úteis na realização das tarefas?

Comentários:

9) De acordo com sua opinião, liste os aspectos positivos da utilização da ferramenta EvoITrack-SocialNetwork.

Comentários:

10) De acordo com sua opinião, liste os aspectos negativos da utilização da ferramenta EvoITrack-SocialNetwork.

Comentários:

11) Você possui alguma sugestão para melhoria da ferramenta EvoITrack-SocialNetwork? Em caso positivo, por favor, especifique-a(s).

() Sim () Não

Comentários:

12) Este espaço é reservado para quaisquer comentários adicionais (dificuldades, críticas e/ou sugestões) a respeito do estudo executado. Contamos com sua contribuição para que o trabalho seja aprimorado.

Comentários:

Novamente, gostaríamos de agradecer pela sua disponibilidade e participação neste estudo.

Andréa Magalhães Magdaleno
Cláudia Maria Lima Werner
Renata Mendes de Araujo