



## EVOLTRACK-PROCESS: UM AMBIENTE PARA A VISUALIZAÇÃO DA COLABORAÇÃO EM PROCESSOS DE SOFTWARE

Francisco Werther Silva de Santana Júnior

Dissertação de Mestrado 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 Mestre em Engenharia de Sistemas e Computação.

Orientadoras: Cláudia Maria Lima Werner  
Andréa Magalhães Magdaleno

Rio de Janeiro  
Abril de 2015

EVOLTRACK-PROCESS: UM AMBIENTE PARA A VISUALIZAÇÃO DA  
COLABORAÇÃO EM PROCESSOS DE SOFTWARE

Francisco Werther Silva de Santana Júnior

DISSERTAÇÃO 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 MESTRE EM  
CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof<sup>a</sup>. Cláudia Maria Lima Werner, D.Sc.

---

Prof<sup>a</sup>. Andréa Magalhães Magdaleno, D.Sc.

---

Prof. Marco Aurélio Gerosa, Ph.D.

---

Prof. Geraldo Bonorino Xexéo, D.Sc.

RIO DE JANEIRO, RJ – BRASIL  
ABRIL DE 2015

Santana Júnior, Francisco Werther Silva de

EvoITrack-Process: Um ambiente para a visualização da colaboração em processos de software / Francisco Werther Silva de Santana Júnior – Rio de Janeiro: UFRJ/COPPE, 2015.

XIV, 126 p.: il.; 29,7 cm.

Orientadoras: Cláudia Maria Lima Werner

Andréa Magalhães Magdaleno

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2015.

Referências Bibliográficas: p. 96-108.

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

## Agradecimentos

À minha família, pelo suporte incondicional que me deram. Foram meus maiores incentivadores. De novo: agradecer pelo que fazem por mim é o mínimo que posso fazer.

À Carina, que não só foi uma das pessoas que mais me ajudou e torceu por mim, mas provavelmente uma das mais importantes pra que tudo isso acontecesse. Não sei como teria sido sem o apoio dela, e sou muito grato por não precisar descobrir.

À minha orientadora, Cláudia Werner, pela disponibilidade, preocupação, oportunidades e pela liberdade que me ofereceu. Foram inúmeras lições durante esse período, as quais eu tenho certeza de que não vou esquecer. Eu só posso agradecer, e espero ter retribuído de alguma forma todo o apoio e confiança que me foram depositados.

À minha coorientadora, Andréa, pelas contribuições, conversas, por me obrigar a fazer cronogramas, por ter tido um papel decisivo para que eu conseguisse terminar o trabalho e, principalmente, por ter me aturado. Essa última missão com certeza não deve ter sido fácil, mas espero que os momentos recompensadores tenham sido mútuos.

Ao pessoal de Belém (filial Rio de Janeiro): Breno, Carol, Chris, David, Junior, Kleverton, Luciana, Mayara, Patricia e Talita. Uns sem dúvida participaram bem mais que outros, mas todos contribuíram no mínimo pra esse número tão grande de pessoas que vieram da terrinha, as vezes fazendo parecer que não saí de lá. Pontos pros churrascos, tacacás, tortas de cupuaçu e até pra lenda do guaraná, que continua sendo só uma promessa há pelo menos uns quatro meses.

Também ao pessoal que tentou me atrapalhar bastante: André, Arthur, Bruno, Felipe, Igor, Jéssica, Juliana, Natália, Ramiro, Roberta, Tarcísio, e vários outros(as). Eles na verdade não contribuíram com absolutamente nada pra esse trabalho, mas seria injusto não mencioná-los já que eles fizeram a vida mais fácil durante esses anos.

Aos funcionários e amigos do LPM/Lab3D, que foram as companhias mais frequentes de trabalho, almoços e etc. São provavelmente as pessoas que mais pensam positivo que já vi, e torço pra que eles tenham tanto sucesso quanto eu sei que torcem pra que eu tenha. E também porque ruim não tá, né?!

Ao pessoal do “grupo de cima”, da LENS-MM, que acabaram participando e contribuindo com o trabalho em apresentações, conversas, confraternizações e afins.

Não posso deixar de fazer um agradecimento especial à Gabriella e ao Humberto por me receberem tão bem em Juiz de Fora, mesmo que por tão pouco tempo!

Finalmente, aos professores Marco Aurélio Gerosa e Geraldo Bonorino Xexéo, que aceitaram o convite para participar desta banca.

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

## EVOLTRACK-PROCESS: UM AMBIENTE PARA A VISUALIZAÇÃO DA COLABORAÇÃO EM PROCESSOS DE SOFTWARE

Francisco Werther Silva de Santana Júnior

Abril/2015

Orientadoras: Cláudia Maria Lima Werner

Andréa Magalhães Magdaleno

Programa: Engenharia de Sistemas e Computação

Esta dissertação de mestrado apresenta um ambiente para entendimento e acompanhamento da colaboração em processos de software. A colaboração é uma característica de grande importância para o desenvolvimento de software, uma vez que projetos de software raramente são frutos de trabalhos individuais, mas sim produtos da combinação de competências e esforços de vários atores que exercem diferentes papéis e executam tarefas interdependentes. A hipótese do trabalho é de que técnicas de visualização de software aplicadas sobre dados de processos podem auxiliar nesse entendimento e acompanhamento da colaboração, facilitando a interpretação de dados relativos à participação dos atores nas instâncias de processo e seus relacionamentos com os demais atores envolvidos em cada execução.

O ambiente concebido, denominado EvolTrack-Process, oferece uma infraestrutura para coleta, transformação e visualização de dados de processos de software. Uma avaliação das técnicas de visualização disponibilizadas em um protótipo do ambiente forneceu indicadores positivos sobre o uso das técnicas projetadas, auxiliando no entendimento e acompanhamento da colaboração em processos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## EVOLTRACK-PROCESS: AN ENVIRONMENT TO VISUALIZE COLLABORATION IN SOFTWARE PROCESSES

Francisco Werther Silva de Santana Júnior

April/2015

Advisors: Cláudia Maria Lima Werner

Andréa Magalhães Magdaleno

Department: Computer Science and Systems Engineering

This dissertation presents an environment to support understanding and monitoring collaboration on software processes. Collaboration is a feature of great importance to software development, since software projects are seldom developed in isolation but normally the product of a joint effort and skills of different actors playing varied roles and performing interdependent tasks. The hypothesis of this study is that software visualization techniques applied to processes execution data can assist in understanding and monitoring collaboration, facilitating the interpretation of data concerning the participation of actors and their interactions with the other actors to perform the activities during the execution of the instances.

The conceived environment, referred to as EvoTrack-Process, offers an infrastructure for software processes data collecting, processing and visualizing. An evaluation of the visualization techniques available in a prototype version of the environment was conducted, providing positive evidence on the usage of the techniques employed in this work.

# SUMÁRIO

1	Introdução.....	1
1.1	Motivação.....	1
1.2	Problema e Questão de Pesquisa .....	2
1.3	Objetivos .....	4
1.4	Metodologia de Pesquisa .....	5
1.5	Organização do Texto .....	7
2	Colaboração em Processos de Software .....	8
2.1	Introdução .....	8
2.2	Colaboração no Desenvolvimento de Software .....	10
2.3	Colaboração sob a Perspectiva de Processos.....	14
2.4	Métricas e Análises de Colaboração .....	16
2.4.1	Medidas de Redes Sociais .....	18
2.5	Considerações Finais .....	22
3	Visualização de Software.....	23
3.1	Introdução .....	23
3.2	Percepção e Cognição no Desenvolvimento de Software .....	24
3.3	Visualizações Aplicadas à Colaboração .....	25
3.4	Visualização no Contexto de Processos.....	31
3.5	Considerações Finais .....	36
4	EvoITrack-Process.....	38
4.1	Introdução .....	38
4.2	Contexto.....	39
4.2.1	COMPOOTIM.....	40
4.3	Visão Geral .....	41
4.4	Extração de Dados.....	43
4.5	Medição da Colaboração.....	46
4.6	Visualizações de Colaboração .....	47
4.7	Considerações Finais .....	50
5	Protótipo .....	51
5.1	Introdução .....	51
5.2	Projeto Técnico .....	52
5.2.1	Requisitos.....	52



5.2.2	Arquitetura .....	53
5.3	Protótipo EvoITrack-Process .....	57
5.3.1	Visualizações e Mecanismos de Interação .....	59
5.3.2	Filtros e Mecanismos de Interação .....	62
5.4	Exemplo de Uso .....	64
5.5	Considerações Finais .....	68
6	Avaliação .....	70
6.1	Introdução .....	70
6.2	Plano de Avaliação .....	70
6.2.1	Objetivo .....	70
6.2.2	Contexto .....	71
6.3	Ameaças à Validade .....	74
6.3.1	Ameaças à validade interna .....	75
6.3.2	Ameaças à validade externa .....	75
6.3.3	Ameaças à validade de conclusão .....	76
6.3.4	Ameaças à validade de construção .....	76
6.4	Execução do Estudo .....	77
6.4.1	Estudo Piloto .....	77
6.4.2	Estudo Observacional .....	78
6.5	Análise dos Resultados .....	83
6.5.2	Análise do Questionário de Avaliação .....	88
6.6	Considerações Finais .....	90
7	Considerações Finais e Trabalhos Futuros .....	91
7.1	Epílogo .....	91
7.2	Contribuições .....	93
7.2.1	Contribuições Tecnológicas .....	93
7.2.2	Contribuições Científicas .....	93
7.3	Limitações .....	94
7.4	Trabalhos Futuros .....	95
	Referências Bibliográficas .....	96
	APÊNDICE A – Dados das Instâncias de Processos Emuladas .....	109
	APÊNDICE B – Instrumentos Utilizados na Avaliação do Ambiente EvoITrack-Process	
	111	
B. 1.	Formulário de Consentimento .....	111

B. 2.	Formulário de Caracterização do Participante .....	113
B. 3.	Formulário para Realização do Estudo.....	116
B. 4.	Questionário de Avaliação do Estudo .....	120
B. 5.	Respostas das Tarefas do Estudo.....	125
ANEXO A – Modelo de Processo Utilizado no Estudo de Avaliação .....		126

# LISTA DE FIGURAS

Figura 1.1 Metodologia de pesquisa adotada .....	5
Figura 2.1. Modificadores sociais para atividades propostos pela notação BPM4People.....	15
Figura 2.2. Exemplo de uma rede social densa (a) e uma esparsa (b) (BARNES, GOLDENA e WASILB, 2012).....	18
Figura 2.3. Exemplo de uma rede social centralizada (a) e descentralizada (b) Fonte: (ALCANTARA, 2010).....	19
Figura 2.4. Exemplo de rede social com os valores de centralidade de cada ator. ....	20
Figura 2.5. Exemplo de rede social, com destaque para o valor de grau de intermediação de um ator equivalente a zero. ....	20
Figura 2.6. Exemplo de rede social, com destaque para o valor de grau de proximidade de um ator. ....	21
Figura 2.7. Exemplos de diferentes medidas de análise de redes sociais aplicadas (adaptado de (ORTIZ-ARROYO, 2010)). ....	21
Figura 3.1. Visão da ferramenta FASTDash. Fonte: (BIEHL, CZERWINSKI, <i>et al.</i> , 2007). ....	26
Figura 3.2. Rede social oferecida pela ferramenta RaisAware, com detalhes para potenciais conflitos entre artefatos. Fonte: (COSTA, FEITOSA e DE SOUZA, 2009). ....	27
Figura 3.3. Uso de visualizações da ferramenta Augur para estudar mudanças nos relacionamentos entre desenvolvedores de software. Fonte: (DE SOUZA, FROEHLICH e DOURISH, 2005).....	28
Figura 3.4. Representação de uma rede sociotécnica na ferramenta EvolTrack- SocialNetwork, com recursos de percepção ativados. Fonte: (VAHIA, MAGDALENO e WERNER, 2011). ....	29
Figura 3.5. Três visualizações exibidas pela ferramenta XFlow, exibindo os relacionamentos dos atores e métricas calculadas sobre a rede social e as dependências entre artefatos. Fonte: (SANTANA, OLIVA, <i>et al.</i> , 2010).....	29
Figura 3.6. Redes sociais de comunicação exibidas pela ferramenta OSSNetwork. Fonte: (BALIEIRO, SOUSA JÚNIOR e DE SOUZA, 2008).....	30
Figura 3.7. Padrões de comunicação encontrados por meio da ferramenta Soyilent. Fonte: (FISHER e DOURISH, 2003).....	31
Figura 3.8. Cidades virtuais em três dimensões utilizadas para retratar a produção de um software (PANAS, BERRIGAN e GRUNDY, 2003). ....	32

Figura 3.9. Múltiplas visões oferecidas em (RILLING, MENG, <i>et al.</i> , 2007) para rastrear solicitações de mudanças em um processo de manutenção. ....	33
Figura 3.10. Eventos exibidos na ferramenta <i>Event Tunnel</i> . ....	33
Figura 3.11. Modelagem colaborativa de processos através de um ambiente de realidade virtual. ....	34
Figura 3.12. Visualização de um <i>workflow</i> ressaltando apenas atividades em que um determinado ator não esteve envolvido. Fonte: (BOBRIK, REICHERT e BAUER, 2007). ....	35
Figura 3.13. Ferramenta Ambrose exibindo um workflow. Fonte: <a href="https://github.com/twitter/ambrose">https://github.com/twitter/ambrose</a> ....	35
Figura 4.1. Visão geral das soluções propostas pelo Projeto CDSOft.....	39
Figura 4.2. Visão geral da COMPOOTIM incluindo o ambiente Evoltrack-Process. ....	40
Figura 4.3. Visão geral do ambiente EvoTrack-Process.....	42
Figura 4.4. Modelo de dados do ambiente EvoTrack-Process. ....	45
Figura 4.5. Exemplo de gráficos exibidos em ferramentas de acompanhamento de processos (KANG e HAN, 2008).....	48
Figura 5.1. Visão geral da arquitetura do ambiente EvoTrack-Process. ....	54
Figura 5.2. Diagrama de classes do módulo de coleta de dados. ....	55
Figura 5.3. Diagrama de pacotes do módulo de visualização. ....	56
Figura 5.4. Tela do protótipo EvoTrack-Process exibindo dados de uma instância de processo. ....	58
Figura 5.5. Tela de coleta de dados do ambiente EvoTrack, exibindo o conector desenvolvido para o EvoTrack-Process. ....	59
Figura 5.6. Visualização de um processo no ambiente EvoTrack-Process.....	60
Figura 5.7. Visualização da rede sociotécnica. ....	61
Figura 5.8. Visualização de métricas. ....	62
Figura 5.9. Filtros de visibilidade disponibilizados no ambiente EvoTrack-Process....	63
Figura 5.10. Filtro de visualização de fluxos mais comuns de processos.....	63
Figura 5.11. Escala de cor utilizada para identificar atividades mais comuns. ....	64
Figura 5.12. Processo de exemplo modelado no Bonita BPMS Studio. ....	65
Figura 5.13. Visualização de uma instância executada do processo.....	66
Figura 5.14. Visualização de uma instância do processo, com destaque para vários atores envolvidos na execução de uma mesma atividade. ....	66
Figura 5.15. Fluxo mais comum de execução do processo.....	67
Figura 5.16. Fluxo mais comum de instâncias onde um ator específico participou. ....	67
Figura 5.17. Visualização de grafo georreferenciado exibindo colaboração entre atores de dois sítios.....	68

Figura 6.1. Formação acadêmica dos participantes do estudo. ....	79
Figura 6.2. Grau de experiência dos participantes do estudo.....	79
Figura 6.3. Tempo de experiência dos participantes do estudo. ....	80
Figura 6.4. Grau de experiência dos participantes do estudo nas áreas do domínio observado.....	81
Figura 6.5. Experiência dos participantes com ambientes de acompanhamento de processos. ....	81
Figura 6.6. Acertos dos participantes do estudo na execução das tarefas. ....	82
Figura 6.7. Acertos dos participantes por tipo de tarefa. ....	82
Figura A.1. Modelo de processos utilizado na avaliação.....	126

## **LISTA DE TABELAS**

Tabela 3.1 Comparativo de abordagens de visualização de processos. ....	36
Tabela 6.1. Definição do objetivo do estudo. ....	71

# 1 Introdução

*Neste capítulo, são expostos a motivação e o contexto de realização da dissertação, sendo também descritos o problema, questão de pesquisa, objetivos e a metodologia de pesquisa adotada. A organização do documento é apresentada ao final do capítulo.*

## 1.1 Motivação

A definição e execução de um processo – independentemente do seu nível de formalidade – pode oferecer maiores chances de sucesso durante o desenvolvimento de produtos de software. Isso é o que relatam vários estudos sobre processos de software<sup>1</sup> (ROZUM, 1993) (FUGGETTA, 2000) (SCHONENBERG, MANS, *et al.*, 2008) e, usando esse argumento como motivação, pesquisadores propuseram uma série de abordagens, técnicas e modelos dedicados a aprimorar processos de software e lidar melhor com seus desafios. Ainda de acordo com essas pesquisas, melhorias na qualidade do processo podem refletir na qualidade do produto desenvolvido, podendo levar à redução no tempo de desenvolvimento e nos recursos gastos com manutenção.

Além de tarefas, artefatos e atores, processos de software também definem outras características fundamentais para o desenvolvimento de sistemas que não são imediatamente reconhecidos em diagramas e modelos, como por exemplo a colaboração<sup>2</sup> entre os envolvidos na sua produção, que surge apenas de forma implícita nas interdependências entre papéis e atividades definidas para o processo.

Uma vez que o desenvolvimento de software não-trivial é um exemplo típico de trabalho colaborativo no qual diferentes atores assumem diversos papéis e executam tarefas interdependentes (WHITEHEAD, 2007), a colaboração é essencial para a produção de software, quer esse desenvolvimento ocorra em um mesmo local geográfico ou em um ambiente distribuído (HERBSLEB, PAULISH e BASS, 2005) (DE SOUZA, MARCZAK e PRIKLADNICKI, 2011). A medida em que sistemas de software

---

<sup>1</sup> Processos de software nesse trabalho são definidos como conjuntos ordenados de tarefas que, quando executadas por atores em uma sequência pre-definida, culminam no desenvolvimento ou evolução de um software, tendo como base as definições de (FEILER e HUMPHREY, 1993) e (SHULL, CARVER e TRAVASSOS, 2001).

<sup>2</sup> Colaboração nesse trabalho é definida como um esforço em conjunto para alcançar um objetivo, conforme (DE VREEDE e BRIGGS, 2005).

se tornam cada vez maiores e mais complexos, podendo ultrapassar milhões de linhas de código, se torna necessário que desenvolvedores de software trabalhem em equipe para construir seus produtos. Segundo DeFranco-Tomarello e Deek (2002), a colaboração na área de Engenharia de Software é, portanto, uma necessidade.

Apesar de sua importância, a colaboração é muitas vezes negligenciada durante a definição do processo, sendo tipicamente considerada como algo que surge naturalmente e não pode ser prevista, muito menos planejada (MAGDALENO, 2013). Esse descaso ou desconhecimento que ocorre quando a definição de processos não leva em consideração as equipes, localidades e capacidades profissionais pode impactar negativamente no desenvolvimento do produto, podendo ocasionar retrabalho, divergências sobre características do produto, sobrecarga e/ou subutilização de atores, entre outros problemas que podem culminar no atraso da entrega do produto, aumento do custo de desenvolvimento e na redução da qualidade do software desenvolvido.

Em relação às ferramentas, embora existam várias para enfrentar as dificuldades e desafios inerentes a processos de software – entre essas soluções, *Process-centered Software Engineering Environments* (PSEEs) (ARBAOUI, DERNIAME, *et al.*, 2014), *Business Process Management Systems* (BPMS) (POURSHAHID, AMYOT, *et al.*, 2007) e ferramentas de *Application Life-Cycle Management* (ALMs) (SCHWABER, 2006) –, a colaboração não é seu foco e são restritas à alocação de atores e informações sobre atividades (e.g. desvios de estimativa, atividades pendentes ou com falhas, entre outras).

## 1.2 Problema e Questão de Pesquisa

No que diz respeito à colaboração, novas práticas para o desenvolvimento de software vem sendo propostas desde o final da década de 90 na tentativa de ampliar a competitividade e oferecer vantagens de mercado, criando uma realidade diferente ao trabalho de algumas equipes e seus atores. O desenvolvimento distribuído de software é um exemplo, no qual atores envolvidos na produção do software estão dispersos geograficamente, o que torna mais fácil a busca por especialistas e oferece um possível aumento de produtividade com o desenvolvimento através de diferentes fusos horários (HERBSLEB e GRINTER, 1999), entre outros benefícios. Outra prática é a chamada *crowdsourcing* (BEGEL, HERBSLEB e STOREY, 2012), onde tarefas são demandadas de forma aberta para que pessoas com ou sem vínculo profissional com o solicitante possam realizá-las, motivadas por recompensas ou até espontaneamente (VON KROGH, HAEFLIGER, *et al.*, 2012). Finalmente, não se pode ignorar duas metodologias de desenvolvimento: os chamados métodos ágeis, que prezam pelo



desenvolvimento de software em um ambiente de colaboração intensa, em geral com times pequenos, reuniões frequentes e com entregas periódicas (DYBA e DINGSOYR, 2009); e o desenvolvimento de software livre, no qual o trabalho colaborativo e voluntário de atores muitas vezes em países diferentes tem originado projetos mundialmente reconhecidos e líderes de mercado (MOCKUS, FIELDING e HERBSLEB, 2000).

Essas práticas e modelos de desenvolvimento, no entanto, introduzem uma série de desafios em relação à comunicação, coordenação e percepção das atividades pelos atores, fazendo com que muitas das empresas que os adotem não consigam alcançar todas essas possíveis vantagens (O'CONCHUIR, HOLMSTROM, *et al.*, 2006) (DYBA e DINGSOYR, 2009): maior produtividade, aproveitamento de capacitações e, no caso específico das metodologias ágeis, redução de problemas de interpretações ambíguas através das reuniões frequentes. Para usufruir dos benefícios que essas práticas oferecem e se manter competitivas, é importante que a indústria saiba como situar e adequar seus processos a essa nova realidade de desenvolvimento – com novos modelos sendo cada vez mais adotados e obtendo grande sucesso – onde os desafios são sobretudo relacionados à colaboração.

A superação desses desafios requer a explicitação de como a colaboração está ocorrendo no processo, para que o problema seja reconhecido e se torne possível a tomada de medidas que favoreçam e estimulem a colaboração. Porém, esse entendimento e acompanhamento de como se deu e/ou está ocorrendo a colaboração entre os atores envolvidos em processos de software não é trivial sem ferramental de apoio dedicado, uma vez que várias instâncias de processos podem ocorrer simultaneamente, com variações em suas definições ou ainda seguindo fluxos diferentes, tornando difícil ter um panorama geral de todos os processos em execução ou que foram finalizados. Além disso, equipes que se tornam indisponíveis durante o desenvolvimento, agentes sobrecarregados de atividades, entre outros, são alguns dos problemas relacionados a colaboração que podem ocorrer e são difíceis de acompanhar sem suporte computacional.

Essas dificuldades motivaram a questão de pesquisa dessa dissertação, elaborada da seguinte forma:

### ***Como entender e acompanhar a colaboração em processos de software?***

Nesse contexto, o conceito de entender desse trabalho está associado diretamente à noção de compreensão apresentada por Bloom (BLOOM, 1984), dizendo respeito à habilidade de captar o significado de um material, possibilitando a interpretação de um material e a elaboração de estimativas e projeções futuras com

base na informação absorvida. O acompanhamento da colaboração, por sua vez, se refere ao entendimento numa escala temporal, ou seja, a capacidade de compreender como as instâncias de processo foram executadas e os modelos de processo evoluíram ao longo do tempo.

Com base nessa questão de pesquisa, a hipótese desse trabalho pôde ser formulada, elaborada da seguinte forma: a visualização de um modelo das instâncias de processo executadas e dos relacionamentos formados quando atores participam de uma mesma tarefa podem auxiliar no entendimento e acompanhamento da colaboração. A hipótese parte da premissa geral da área de Visualização de Software, de que técnicas de visualização podem auxiliar na identificação de padrões e interpretação de dados de processos de software, e sua utilização poderia, portanto, ajudar empresas que desenvolvem software a entender, gerenciar e evoluir seus processos também sob essa perspectiva de colaboração.

Essa pesquisa segue o paradigma *Design Science* (HEVNER, MARCH, *et al.*, 2004), que propõe a criação de um artefato tecnológico para um problema de um domínio específico, nesse caso o entendimento e acompanhamento da colaboração em processos de software. Ainda de acordo com o método, o artefato gerado deve ser inovador, solucionando um problema ainda não resolvido ou apresentando uma solução mais eficaz e/ou eficiente, devendo ser posteriormente avaliado e comunicados o cenário de utilização do artefato e seus potenciais benefícios, apresentados nos próximos capítulos.

### **1.3 Objetivos**

O objetivo dessa dissertação de mestrado é construir um ambiente para visualizar e monitorar a colaboração em processos de software, visando apoiar o entendimento e acompanhamento da colaboração que está ocorrendo ou ocorreu durante o desenvolvimento de software, e, eventualmente, possibilitar o estímulo a colaboração a partir da visualização de execuções de processos em andamento ou com base no conhecimento adquirido ao observar instâncias previamente executadas.

De modo a alcançar o objetivo principal desse trabalho, foram estabelecidos os seguintes objetivos específicos:

- Levantar características que possam apoiar o entendimento e acompanhamento da colaboração em processos de software;
- Especificar e desenvolver um ambiente que possibilite a coleta de dados de processo e apresente informações sobre colaboração, tanto para instâncias em andamento quanto para aquelas já finalizadas;

- Avaliar o ambiente proposto, de modo a verificar se ele realmente é capaz de auxiliar no entendimento e acompanhamento da colaboração em processos.

## 1.4 Metodologia de Pesquisa

Os objetivos específicos da dissertação foram alcançados através da seguinte metodologia (Figura 1.1):

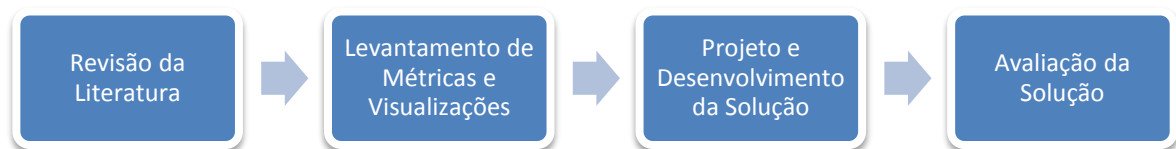


Figura 1.1 Metodologia de pesquisa adotada

**Revisão da literatura.** A revisão da literatura foi feita de forma *ad-hoc*, buscando por artigos de colaboração em processos de software. Uma pesquisa inicial nesse tema retornou um número muito grande de resultados, com diversos estudos que reforçavam a importância e desafios da colaboração em geral (VOGELSANG e CARSTENSEN, 2001) (WHITEHEAD, 2007); atividades, metamodelos e características de modelos de processos de desenvolvimento de software (KUSUMASARI, SUPRIANA, *et al.*, 2011) (KEDJI, LBATH, *et al.*, 2012) (FAN, LI e ZHAO, 2012); ferramentas de *groupware* (DEFRANCO-TOMMARELLO e DEEK, 2002) (LEI, CHAKRABORTY, *et al.*, 2004) (ROSENTHAL e FINGER, 2006), entre outros subtópicos, que contribuíram para a base desse trabalho.

O volume muito grande de resultados dispersos acabou direcionando a busca especificamente para a visualização da colaboração em processos, e neste contexto não foi identificado outro trabalho com foco em monitoração da colaboração em processos, apenas outras ferramentas que tentavam facilitar a execução de uma atividade de forma colaborativa entre times, por meio de alguma solução: *wikis*, onde membros de uma organização podem contribuir para a solução de uma atividade ou construção de um documento de forma colaborativa (LICCARDI, DAVIS e WHITE, 2008), ou tecnologias que utilizam dispositivos se comunicando de forma síncrona e assíncrona, permitindo que atores executem tarefas simultaneamente (BALOIAN, ZURITA, *et al.*, 2011), entre outras.

**Levantamento de métricas e visualizações de processos.** Após a busca por artigos de colaboração em processos, uma outra revisão *ad-hoc* mais específica foi realizada, com foco em métricas de colaboração e técnicas de visualização aplicadas à processos. A busca por métricas de colaboração surge da necessidade de capturar dados que consigam expressar essa característica em processos de software, para

servir como base para que os usuários do ambiente entendam como a colaboração ocorreu e possam acompanhá-la. O foco em técnicas de visualizações de processo, por sua vez, teve o objetivo de levantar metáforas e técnicas de visualização já aplicadas à processos que pudessem também ser adequadas sob a perspectiva de colaboração tratada nessa dissertação.

Com um número muito pequeno de trabalhos retornados, as referências dos artigos e os anais das conferências onde foram publicados também foram levados em consideração, o que incluiu as conferências: International Software Metrics Symposium, International Conference on Information Visualization (IV), IEEE Information Visualization Conference (InfoVis), IEEE Visual Analytics Science and Technology (VAST), além do Congresso Brasileiro de Software (CBSOFT).

Não foi identificada nenhuma métrica criada especificamente para medir a colaboração em processos de software. Essa ameaça foi contornada com o uso de métricas de grafos, encontradas com frequência em trabalhos que adotavam a análise de redes sociais para inferir colaboração durante o desenvolvimento de software (MADEY, FREEH e TYNAN, 2002) (CROSS, BORGATTI e PARKER, 2002) (SANTOS, ARAUJO e MAGDALENO, 2011) (COSTA, SANTANA e DE SOUZA, 2009).

A respeito de visualizações em processos, foram encontrados poucos trabalhos, incluindo tanto propostas que desenvolveram suas próprias metáforas visuais para destacar alguma característica dos processos, como a metáfora *Event Tunnel* (SUNTINGER, OBWEGER, *et al.*, 2007) quanto outras que adotaram técnicas já definidas e utilizadas pela academia em outros contextos, como o uso de cidades virtuais e seus prédios para a representação de características do desenvolvimento de software (PANAS, BERRIGAN e GRUNDY, 2003).

**Projeto e desenvolvimento da solução.** Com base no conhecimento sobre técnicas e propostas similares adquirido nas etapas anteriores do trabalho, foi dado início ao projeto e desenvolvimento de um ambiente que pudesse lidar com o problema abordado nessa dissertação. Assim, teve origem o *EvoTrack-Process* (SANTANA, 2014), ambiente que oferece visualizações para evidenciar a colaboração em processos de software e acrescenta um novo potencial de análise para o projeto *EvoTrack* (WERNER, MURTA, *et al.*, 2011).

**Avaliação da solução.** Como última etapa, foi planejado e conduzido um estudo experimental para verificar a viabilidade e aplicabilidade das visualizações planejadas para o ambiente para auxiliar no entendimento e acompanhamento da colaboração em processos de software. Este estudo teve como participantes alunos e profissionais vinculados à academia, e os seus resultados indicaram que as técnicas de visualização aplicadas nesse trabalho podem ser um caminho viável para auxiliar

no entendimento e acompanhamento de características associadas à colaboração em processos de desenvolvimento de software.

## 1.5 Organização do Texto

Este primeiro capítulo, **Introdução**, apresentou o contexto no qual essa dissertação está inserida, problema, questão de pesquisa, hipótese, objetivos e metodologia de pesquisa definida. O restante dessa dissertação está organizado como segue:

O Capítulo 2 trata do tema central dessa dissertação: a **Colaboração em Processos de Software**. São discutidos trabalhos e propostas sobre esse tema, assim como formas de medir a colaboração em processos de software.

O Capítulo 3 discute a área de **Visualização de Software**, explicando conceitos e técnicas importantes, bem como de que formas a área pode e vem sendo empregada no contexto de processos de software.

No Capítulo 4, o ambiente proposto nessa dissertação, **EvoITrack-Process**, é apresentado, sendo descritos seu contexto e etapas de funcionamento.

No Capítulo 5, um **Protótipo** que atende as necessidades estabelecidas pelo ambiente e incorpora as técnicas escolhidas é exposto, sendo descritos seus princípios de desenvolvimento, decisões de projeto, funcionalidades, e exemplos de como o ambiente pode ser utilizado.

O Capítulo 6 discute a **Avaliação** realizada sobre o ambiente proposto, apresentando o planejamento do estudo, detalhando sua execução e, por fim, seus resultados e ameaças à validade.

Finalmente, o Capítulo 7, encerra com as **Considerações Finais e Trabalhos Futuros** dessa dissertação, onde são ressaltadas as contribuições e limitações do trabalho e discutidas oportunidades para trabalhos futuros.

## 2 Colaboração em Processos de Software

*Esse capítulo discute a importância da colaboração para o desenvolvimento de software, apresentando benefícios, desafios e tecnologias criadas para lidar com esse aspecto. A colaboração sob a perspectiva de processos de software é discutida em mais detalhes, assim como métricas e técnicas para analisá-la.*

### 2.1 Introdução

A colaboração é alvo de diversos estudos em várias áreas de conhecimento, sobretudo no campo das ciências sociais. No entanto, embora seja um tema muito pesquisado, não há consenso sobre o termo, e JOHN-STEINER *et al.* (1998) tratam a própria definição e o escopo relacionado à colaboração como um dos desafios das pesquisas no tema. Essa dissertação segue a definição de DE VREEDE e BRIGGS (2005), que definem colaboração como “um esforço em conjunto para alcançar um objetivo”.

Dentro da área de colaboração, pesquisadores investigam vários sub-temas. ELLIS, GIBBS e REIN (1991) propuseram três áreas fundamentais para entender e auxiliar a interação entre pessoas e o trabalho em grupo: a comunicação, colaboração e a coordenação. Esse trabalho embasou a criação do Modelo 3C de Colaboração (PIMENTEL, GEROSA, *et al.*, 2006), envolvendo as mesmas três áreas: a comunicação através de troca de mensagens; a coordenação, através do gerenciamento de pessoas, atividades e recursos; e a cooperação através de um espaço compartilhado para execução das tarefas.

Seguindo um raciocínio similar, ARAUJO e BORGES (2007) identificaram quatro características que apoiam a colaboração: comunicação, coordenação, percepção e memória de grupo. De acordo com os autores, para colaborar, pessoas precisam trocar informações (comunicação), organizar seus trabalhos (coordenação), atuar juntos em espaços de trabalho coletivos (memória de grupo) e ter percepção do que está acontecendo para obter informações necessárias para realização das suas atividades.

No campo da Engenharia de Software, é possível afirmar que a colaboração permeia o desenvolvimento de software. Raramente a construção de um software atual ocorre de forma individual, mas sim como resultado de um processo intensivamente colaborativo sobre o qual várias pessoas assumem diferentes papéis,

se comunicando, coordenando e interagindo umas com as outras (VALETTO, HELANDER, *et al.*, 2007) (DE SOUZA, MARCZAK e PRIKLADNICKI, 2011).

Através da colaboração, atores envolvidos no desenvolvimento de software podem produzir resultados potencialmente melhores do que trabalhando de forma individual (FUKE, RAPOSO, *et al.*, 2002). O poder da tomada de decisões, a capacidade de analisar alternativas tecnológicas e a complementação de capacidades e de esforços individuais são algumas das características que favorecem o trabalho colaborativo diante do individual (TUROFF e HILTZ, 1982) (HAINES e VEHRING, 2012). No ambiente educacional, alguns estudos demonstraram que a realização de tarefas de forma colaborativa aprimora a capacidade dos estudantes em solucionar problemas no contexto da Engenharia de Software (NOSEK, 1998) (DEFRANCO-TOMMARELLO e DEEK, 2002). Além dessas vantagens, a colaboração, quando bem sucedida, aumenta também a satisfação das pessoas, que se sentem mais envolvidas com o projeto e a organização, promove a inovação e o aprendizado entre times (DEMARCO e LISTER, 1999) (PAULUS e NIJSTAD, 2003).

No entanto, estimular a colaboração no contexto de software não é tão simples quanto o adágio “duas cabeças pensam melhores que uma”. Quando a colaboração ocorre de forma ineficiente durante o desenvolvimento de software, uma série de problemas se tornam mais prováveis de acontecer: resistência para ouvir ideias ou colaborar com pessoas fora de círculos de confiança (conhecida como a Síndrome do Não Inventado Aqui), diminuição de produtividade e sobrecarga de atores, entre outros (KATZ e ALLEN, 1982) (PAASIVAARA e LASSENIUS, 2003). A medida em que mais pessoas precisam trabalhar em conjunto, aumenta também a dificuldade em acompanhar as tarefas de cada membro da equipe e, conseqüentemente, o risco de atrapalhar ou duplicar o trabalho de outros (BROOKS, 1978). Quando o desenvolvimento de software utiliza agentes em mais de um local geográfico, os desafios relacionados à colaboração são ampliados ainda mais, surgindo barreiras temporais, culturais, etc. (HERBSLEB, MOCKUS, *et al.*, 2001).

Diante da necessidade de colaboração entre os indivíduos que participam do desenvolvimento de software e dos riscos que a colaboração pode oferecer, se tornam necessários mecanismos que possibilitem que ela seja planejada e acompanhada (MAGDALENO, 2013). O entendimento de como as pessoas se organizam e interagem umas com as outras pode ser determinante para aprimorar a produtividade e a qualidade dos produtos desenvolvidos (VALETTO, HELANDER, *et al.*, 2007).

Nesse sentido, um dos maiores desafios para alcançar os benefícios e superar as dificuldades relacionadas à colaboração reside em como explicitá-la, de modo que os atores envolvidos no desenvolvimento de um projeto percebam como suas

atividades se relacionam com o trabalho dos demais atores (ARAUJO e BORGES, 2007).

Essa explicitação da colaboração pode ocorrer através dos processos da organização. Em suas definições, os processos de software trazem as atividades, sequências de tarefas e elementos que ditam como os agentes precisam se organizar para cumprir os seus objetivos de desenvolvimento, e dessa forma interferem diretamente na colaboração necessária para a criação de um produto. Sendo assim, o aumento da visibilidade da colaboração em processos, sobretudo aqueles mais complexos e com ciclos de vida mais longos, é de grande importância para o reconhecimento de problemas e o planejamento de ações para contorná-los (MOHTASHAMI, MARLOWE e KU, 2011).

Este capítulo apresenta um panorama sobre trabalhos e técnicas que lidam com colaboração no desenvolvimento de software, e está organizado como segue: a Seção 2.3 apresenta alguns modelos de desenvolvimento de software que estressam a colaboração entre os atores do desenvolvimento, incluindo alguns que se beneficiariam mais das ideias apresentadas nessa dissertação; a Seção 2.4 discute a colaboração sob uma perspectiva específica de processos, apresentando trabalhos com objetivos mais alinhados ao dessa dissertação; a Seção 2.4 discorre sobre métricas e mecanismos para análises de colaboração, bem como de que forma técnicas de análise de redes sociais podem ser utilizadas com essa finalidade; finalmente, a Seção 0 apresenta as considerações finais do capítulo.

## **2.2 Colaboração no Desenvolvimento de Software**

Como mencionado anteriormente, a Engenharia de Software é um campo essencialmente colaborativo para qualquer projeto que não seja excessivamente simples e/ou pequeno. Contudo, o grau de colaboração exigido pode variar significativamente entre diferentes contextos de desenvolvimento (HERBSLEB, MOCKUS, *et al.*, 2001).

Nas últimas décadas, os cenários de desenvolvimento de software têm apresentado diferenças radicais entre si, e variam desde uma ou mais equipes colocalizadas dentro de um mesmo prédio; equipes dispersas geograficamente em vários sítios de uma empresa; até a interação entre equipes de organizações distintas, colaborando para o desenvolvimento ou estabelecimento de uma plataforma comum de negócios. Além disso, de acordo com o tipo de software que precisa ser desenvolvido e dos objetivos organizacionais, determinados modelos de



desenvolvimento de software<sup>3</sup> podem ser mais ou menos indicados que outros, não existindo, portanto, uma forma única adequada para o desenvolvimento de qualquer tipo de software (HUMPHREY, 1989).

Assim como não há uma forma única ideal para desenvolver software, do ponto de vista da colaboração não existem práticas e planejamentos que garantam que ela sempre vá ocorrer de forma efetiva. A própria estrutura da organização tem um papel fundamental e não pode ser negligenciada durante o planejamento de ambos o processo e arquitetura do produto a ser desenvolvido (CONWAY, 1968) (PARNAS, 1972), com alguns modelos e contextos de desenvolvimento sofrendo mais ou menos impacto decorrentes da colaboração.

Dentre os modelos de desenvolvimento, um tema que tem tido muito destaque desde a publicação de seu manifesto em 2001 são os chamados métodos ágeis (AGILE ALLIANCE, 2001). Os métodos ágeis foram concebidos por um grupo experiente de engenheiros de software como uma alternativa para resistir melhor aos desafios que o mercado de software impõe à indústria de software, estabelecendo um ritmo acelerado de avanços tecnológicos e de mudanças em demandas de clientes. Algumas das ideias por trás do desenvolvimento ágil são fundamentalmente colaborativas, tais como a redução do tamanho das equipes, a valorização da proximidade física e intenso diálogo entre os desenvolvedores, e práticas de trabalho colaborativas, como o *pair programming*. Dentre as vantagens dos métodos ágeis, estudos apontam para um aumento na qualidade de código construído e na produtividade, necessitando para isso um comprometimento da alta gerência de software para garantir a utilização das práticas ágeis e de membros de equipes mais experientes (LINDVALL, BASILI, *et al.*, 2002) (DYBA e DINGSOYR, 2009).

O destaque que os métodos ágeis têm recebido nos últimos anos tanto da indústria quanto pela comunidade acadêmica tem servido como motivação para aumentar a consciência da importância da colaboração para o desenvolvimento de software. A ênfase em colaboração e comunicação constante defendida por essas metodologias estimula a investigação de desafios relacionados a colaboração e como eles podem ser contornados e do estudo de práticas colaborativas e seus efeitos.

Em relação aos diferentes contextos de desenvolvimento, o uso de atores espalhados em diversas localidades geograficamente dispersas tem se tornado uma norma para empresas envolvidas com software (HERBSLEB, MOCKUS, *et al.*, 2001)

---

<sup>3</sup> Modelos de desenvolvimento de software oferecem definições em alto nível de abstração de um conjunto de práticas e recomendações para o desenvolvimento de um software (SOMMERVILLE, 2004).

(RAMASUBBU, CATALDO, *et al.*, 2011). Essas companhias buscam vantagens que o desenvolvimento de software em uma escala global pode oferecer, sendo algumas delas: o acesso a recursos humanos mais especializados e/ou mais baratos, que reduzem o custo de desenvolvimento do produto; o aumento da produtividade através do desenvolvimento contínuo, onde desenvolvedores em regiões temporais diferentes mantêm o desenvolvimento em até 24 horas por dia; e o aumento da proximidade com consumidores e mercados locais (ÅGERFALK, FITZGERALD, *et al.*, 2008).

No entanto, a literatura acadêmica também é rica de casos de insucessos relacionados ao desenvolvimento distribuído e/ou global de software, onde a grande maioria dos casos não consegue superar desafios que são relacionados principalmente à colaboração. Embora as vantagens do desenvolvimento global de software sejam tentadoras, alcançá-las é extremamente difícil, haja visto que é necessário lidar com problemas estratégicos, técnicos e culturais de times em regiões temporais diferentes e afastados fisicamente (HERBSLEB e GRINTER, 1999) (HERBSLEB, MOCKUS, *et al.*, 2001). Um estudo em uma empresa com quatro sítios em dois continentes diferentes mostrou que desenvolvedores em contato com membros em localidades diferentes precisavam esperar em média duas vezes e meia mais tempo para obter informações críticas para a realização de suas tarefas que aqueles que dependiam apenas de membros do seu próprio sítio (HERBSLEB, MOCKUS, *et al.*, 2001).

Existe ainda outro modelo de desenvolvimento que também requer colaboração intensa entre os envolvidos: o desenvolvimento de software livre (RAYMOND, 2001) (FSF, 2008). Software livres são frutos do trabalho em equipe de pessoas que muitas vezes não possuem qualquer tipo de vínculo entre si, mas mesmo assim são capazes de se coordenar e construir um produto. Com vários casos de sucesso, incluindo o sistema operacional Linux, um grande competidor no mercado de sistemas operacionais, e o servidor Apache, alcançando aproximadamente 70% do seu nicho de mercado (NETCRAFT, 2005), o desenvolvimento de software livre é reconhecido como um modelo de desenvolvimento capaz de produzir software de alta qualidade (MOCKUS, FIELDING e HERBSLEB, 2002).

Assim como o desenvolvimento distribuído de software, o modelo de desenvolvimento de software livre também pode sofrer dificuldades relacionadas à colaboração. O ato de contribuir e se coordenar com outras pessoas é uma tarefa extremamente complexa, e faz com que várias pessoas interessadas em participar em projetos de software livre desistam por barreiras de linguagem ou culturais, entre outras (STEINMACHER, CHAVES, *et al.*, 2014). A dificuldade de coordenação entre seus colaboradores também é um problema, e foi responsável por um hiato de

aproximadamente vinte meses de desenvolvimento no projeto GIMP quando seus líderes abandonaram o projeto e não haviam outros colaboradores capazes de dar continuidade ao desenvolvimento (YE e KISHIDA, 2003).

Além desses casos em que um software é desenvolvido por companhias ou grupos para atender interesses particulares ou de uma comunidade específica, a colaboração no desenvolvimento de software também pode ser observada em nível interorganizacional. Todo dispositivo ou software é sempre elaborado com contribuições de outros fornecedores, se sobrepondo, interligando ou interagindo com outros sistemas e estabelecendo relacionamentos entre fornecedores e produtores de TI distintos (LUNGU, LANZA, *et al.*, 2010), que, mesmo quando concorrentes, atuam juntos para o estabelecimento de uma plataforma comum de software, dando origem aos chamados ecossistemas de software (MESSERSCHMITT, 2003).

Ecossistemas de software podem ser definidos como “um conjunto de atores funcionando como uma unidade e interagindo com um mercado compartilhado de software e serviços” (JANSEN, BRINKKEMPER e FINKELSTEIN, 2009). Um exemplo de ecossistema de software é o ambiente de desenvolvimento Eclipse, onde vários membros de comunidades e grupos da indústria atuam sobre uma mesma plataforma comum, evoluindo-a e criando uma série de extensões para o ambiente.

No que diz respeito à colaboração, diversas pesquisas em ecossistemas de software estudam aspectos fundamentalmente colaborativos, como CATALDO e HERBSLEB (2010) que apresentam o conceito de translucidez, uma característica crucial para a coordenação e a colaboração entre desenvolvedores de aplicações de um ecossistema, correspondendo a quais informações e como o projeto técnico da plataforma deve ser disponibilizado para a comunidade que trabalhará sobre a própria plataforma, ou seja, o quão translucidos são os comportamentos e as informações de um ecossistema.

Em relação às ferramentas de apoio, em todos os cenários descritos anteriormente, até mesmo para manutenção de ecossistemas de software (CATALDO e HERBSLEB, 2010), é comum o uso de tecnologias colaborativas que facilitem o trabalho das equipes durante o desenvolvimento de software. Essas ferramentas, denominadas ferramentas de *groupware*<sup>4</sup>, tentam conectar os desenvolvedores através do uso de tecnologias, auxiliando-os a superar dificuldades de comunicação, coordenação e cooperação (FUKS, RAPOSO, *et al.*, 2002), e variam desde

---

<sup>4</sup> *Groupware* pode ser definido como “qualquer ferramenta de software que auxilia ou amplia o trabalho em equipe” (GREENBERG, 1991).

repositórios compartilhados, estabelecimento de canais de comunicação *online* e *offline*, e mecanismos que ampliem a percepção das tarefas de cada indivíduo (MANGAN, BORGES e WERNER, 2004).

Embora as ferramentas de *groupware* tenham um papel fundamental para auxiliar desenvolvedores a lidar com eventuais barreiras temporais, geográficas e culturais, é também importante a criação de tecnologias voltadas para aqueles responsáveis por gerenciar o desenvolvimento de software. Para esses atores associados à gerência, a visibilidade da colaboração em um nível organizacional pode ser ampliada através de mecanismos que ofereçam informações sobre como a colaboração ocorre em processos de software (MAGDALENO, 2013).

## 2.3 Colaboração sob a Perspectiva de Processos

Além dos estudos e tecnologias propostas para facilitar a comunicação, coordenação e realização de algumas tarefas associadas ao desenvolvimento de software, a colaboração também é foco de estudos e propostas sob uma perspectiva específica de processos.

A abstração de processos permite um tratamento diferente da colaboração em relação aos trabalhos apresentados anteriormente, definindo tarefas, notações e modificações de processos para incorporar iniciativas colaborativas; na manipulação e manutenção de informações que compõem a memória da organização de uma forma colaborativa e na avaliação do nível de colaboração em uma organização, por meio de modelos de maturidade criados especificamente com essa finalidade.

No contexto de tarefas e extensões de capacidades de processos que lidam mais explicitamente com a colaboração, existem iniciativas como o *Collaborative Software Process* (CSP) e o BPM4People. O CSP (WILLIAMS, 2000) é um modelo de processo composto por uma série de formulários, *checklists* e atividades adaptadas para a prática de desenvolvimento em pares, e visa ampliar a percepção de desenvolvedores e facilitar o trabalho em equipe. Um estudo demonstrou que estudantes de computação conseguiram desempenhar suas tarefas em um tempo muito menor e com maior qualidade utilizando o processo CSP e suas práticas colaborativas (WILLIAMS, 2000).

Voltada para processos de negócios, a iniciativa BPM4People (BRAMBILLA, FRATERNALI e RUIZ, 2012) propôs uma extensão da notação *Business Process Model and Notation* (BPMN), e incluiu novas representações de caráter colaborativo que expandem as capacidades de representação e execução de um processo. Usando o BPM4People é possível, por exemplo, criar atividades onde os atores que as realizarão serão convidados durante a execução do processo (Figura 2.1).





Comment	 Comment
Vote	 Vote
Invite	 Invite
Rank	 Rank

Figura 2.1. Modificadores sociais para atividades propostos pela notação BPM4People.

A colaboração também aparece associada aos processos sob uma perspectiva de memória organizacional. De acordo com a teoria da criação do conhecimento organizacional, organizações que lidam com inovação não absorvem apenas informações externas para criar soluções, mas também criam novos conhecimentos e abordam problemas com base em suas informações internas (NONAKA e KONNO, 1998). Nesse sentido, a colaboração surge como meio para suportar a melhoria de processos, atuando na socialização, manutenção e institucionalização das práticas dos processos da organização (BANNON e KUUTTI, 1996) (ARENT e NORBJERG, 2000).

Em outra linha de trabalhos, alguns pesquisadores propuseram modelos de maturidade e sistemáticas voltados explicitamente para colaboração. Hain e Back (2009) encontraram 55 modelos sob esse tema, incluindo tanto propostas acadêmicas quanto práticas. Os modelos de maturidade são modelos de referência que atuam como um roteiro, definindo um conjunto ordenado de práticas para a melhoria incremental, comparação ou avaliação da situação atual das organizações, apresentando etapas e guias para colocar a empresa em um caminho evolutivo, partindo de um processo caótico *ad-hoc* até um ambiente controlado e mais disciplinado. Uma descrição e análise comparativa entre modelos de maturidade criados especificamente para colaboração pode ser obtida em (MAGDALENO, ARAUJO e WERNER, 2011).

É importante notar que esses modelos de processo e maturidade reconhecem que algumas de suas práticas são difíceis de cumprir sem o apoio de ferramentas apropriadas, reforçando a importância de iniciativas como o ambiente proposto nessa dissertação.

Com base nos estudos encontrados e na importância da colaboração, foi observado que embora a colaboração seja de fato um tema de grande relevância no

contexto de processos de software, poucas abordagens existem para dar suporte no **entendimento** e **acompanhamento** da colaboração em processos. Essa percepção motiva o desenvolvimento de um novo ambiente que possa oferecer esses mecanismos para gerentes de software.

## 2.4 Métricas e Análises de Colaboração

A medida em que organizações de software crescem e seus processos de desenvolvimento se tornam mais complexos, a organização passa a depender cada vez mais de relatórios e programas de medição para acompanhar os produtos que estão em desenvolvimento. Essas medidas são fundamentais para que a empresa possa estimar custos e prazos, acompanhar defeitos, e estabelecer uma *baseline* para comparar o andamento de desenvolvimentos futuros (MOORE, 1999) (CLARET, 2013).

Na Engenharia de Software, o campo de métricas é vasto e inclui medidas criadas com diversas finalidades: avaliar o estado do projeto, estimar confiabilidade de produtos, medir o desempenho de atores, prover informações para auditores e autoridades externas (KANER e BOND, 2004). No entanto, a revisão da literatura realizada corrobora com a afirmação de que existe uma lacuna a respeito de métricas voltadas especificamente para colaboração, tanto em processos de desenvolvimento software como também em processos de negócio (THOMSON, PERRY e MILLER, 2007) (MOHTASHAMI, MARLOWE e KU, 2011) (CLARET, 2013). Esse é um problema grave para estudos de colaboração, uma vez que a utilização de métricas sem entendimento do que elas indicam pode levar a distorções na interpretação dos valores (KANER e BOND, 2004).

Nesse contexto, vários pesquisadores têm recorrido a análise de redes sociais<sup>5</sup> para observar a colaboração em diversas estruturas sociais. Apenas em 2008, o índice de citação Science Citation Index<sup>6</sup> apontou 1269 artigos relacionados a redes sociais (KADUSHIN, 2012). Em geral, esses trabalhos recorrem as redes sociais devido a sua capacidade em evidenciar relacionamentos entre um grupo de atores, incluindo os padrões e implicações desses relacionamentos, possibilitando responder questões de

---

<sup>5</sup> Redes sociais podem ser definidas como uma rede de relacionamentos interligando entidades sociais. A análise de redes sociais é um campo multidisciplinar de pesquisa que engloba modelos formais e teorias com fundamento matemático e estatístico com teorias sociais (WASSERMAN e FAUST, 1994).

<sup>6</sup> O índice SCI agrega uma série de periódicos de diversas áreas de conhecimento. Mais informações podem ser obtidas em: <http://thomsonreuters.com/science-citation-index-expanded>.

ciências sociais através de definições formais precisas para aspectos políticos, econômicos ou sociais (WASSERMAN e FAUST, 1994). WHITE, BOORMAN e BREIGER (1976) chegam a afirmar em um trabalho que influenciou muitas outras pesquisas que “conceitos de redes podem oferecer a única forma de construir uma teoria de estruturas sociais”.

Exemplos de redes sociais podem ser encontradas em todos os lugares, como entre estudantes em salas de aula, funcionários que trabalham juntos, em colaborações científicas através da coautoria de artigos acadêmicos, entre outros (BARABASI e BONABEAU, 2003). WASSERMAN e FAUST (1994) apresentam uma lista de tópicos nos quais redes sociais foram aplicadas para auxiliar a descoberta de propriedades de estruturas sociais, incluindo a resolução de problemas em grupo e relações políticas e econômicas mundiais.

Utilizando redes sociais aplicadas ao desenvolvimento de software, pesquisadores já foram capazes de investigar características arquiteturais de software (MACCORMACK, RUSNAK e BALDWIN, 2006); identificar grupos mais importantes e acompanhar a evolução de desenvolvedores (DE SOUZA, FROEHLICH e DOURISH, 2005) (OLIVA, SANTANA, *et al.*, 2012), entre outros temas, fazendo com que a área de análise de redes sociais fosse vista como uma alternativa promissora para investigação de aspectos do desenvolvimento de software.

Nesse cenário, as redes sociais têm sido utilizadas sobretudo na investigação de fenômenos relacionados ao desenvolvimento de software livre, onde a disponibilidade de dados e quantidade de relacionamentos passíveis de serem observados não é um problema. Um exemplo dessas pesquisas foi conduzida por MADEY, FREEH e TYNAN (2002), que modelaram o desenvolvimento de mais de 39 mil projetos software livre envolvendo mais de 33 mil desenvolvedores como uma rede social colaborativa, onde foram constituídos relacionamentos entre atores que participaram de um mesmo projeto. Os pesquisadores constataram que a colaboração entre desenvolvedores de software não segue a mesma distribuição de redes de outros domínios profissionais, como o domínio da indústria de cinema; que aproximadamente 80% de desenvolvedores contribuíram para apenas um projeto; e outros achados. TRAINER *et al.* (2005), por sua vez, utilizaram redes sociais que mapeavam as dependências técnicas de um projeto de software, ampliando a percepção de desenvolvedores para eventuais propagações de defeitos que poderiam surgir ao modificar um artefato e possibilitando a identificação de com quem cada desenvolvedor deveria se comunicar e colaborar para a coordenação de suas tarefas.

Diversos trabalhos que aplicam redes sociais utilizam propriedades das redes para condução de suas pesquisas. Essas propriedades oferecem diversos indicadores

que evidenciam características específicas da estrutura social construída, e algumas dessas propriedades foram selecionadas e disponibilizadas nesse trabalho com o intuito de oferecer medidas para auxiliar no entendimento da colaboração em processos de software, descritas na subseção a seguir.

### 2.4.1 Medidas de Redes Sociais

Os relacionamentos e a estrutura geral das redes sociais podem oferecer indicadores ricos do relacionamento de um grupo e como essas pessoas interagem. Duas medidas de redes sociais que podem dar indícios sobre a colaboração entre os atores são os valores de densidade e centralização (WASSERMAN e FAUST, 1994).

A **densidade** de uma rede social é um valor correspondente à razão entre o número de suas arestas e o número de arestas máximo que a rede poderia ter, variando de 0 à 1. O cálculo da densidade  $D$  de uma rede obedece à expressão abaixo, em que  $A$  equivale ao número de arestas da rede e  $V$  ao número de vértices.

$$D = \frac{2A}{V(V - 1)}$$

A densidade é, então, um indicador geral dos relacionamentos da rede (Figura 2.2). Quando o valor da densidade equivale a 1, todos os atores da rede têm algum tipo de relacionamento entre si; e quando o valor é 0, não há nenhum tipo de relacionamento entre quaisquer atores da rede. No que diz respeito à redes de colaboração, o valor de densidade pode ser utilizado para entender o nível de coordenação entre os atores. Nessas redes, valores baixos de densidade podem indicar a existência de líderes que monopolizam relacionamentos e permeiam a relação com os demais membros, enquanto valores altos de centralidade podem indicar uma maior troca de informações e uma coordenação distribuída entre os membros da rede.

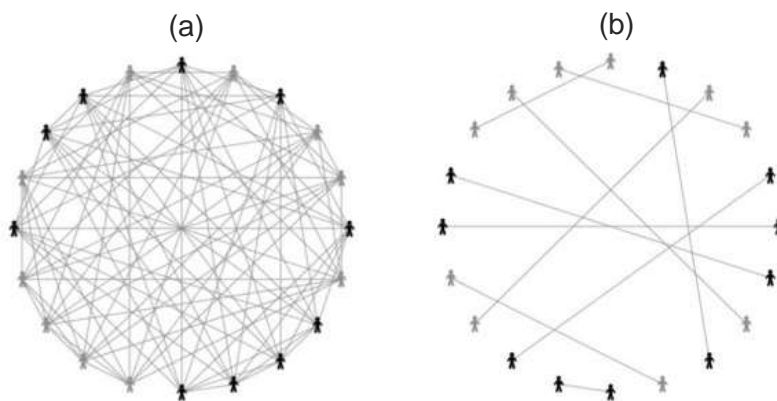


Figura 2.2. Exemplo de uma rede social densa (a) e uma esparsa (b) (BARNES, GOLDENA e WASILB, 2012).



A **centralização**, por sua vez, é uma medida que indica quão central é a posição do nó de maior centralidade de uma rede em relação a centralidade dos outros membros.

O cálculo da centralização leva em conta, portanto: a) a soma de todas as diferenças ( $P_*$ ) entre o maior valor de centralidade da rede e os demais (representado por  $C_x(P_*)$ ); b) a média da maior diferença entre as centralidades dos nós (representado por  $\max \sum_{i=1}^V C_x(P_*) - C_x(P_i)$ ), obedecendo a fórmula abaixo (FREEMAN, 1979).

$$C = \frac{\sum_{i=1}^V C_x(P_*) - C_x(P_i)}{\max \sum_{i=1}^V C_x(P_*) - C_x(P_i)}$$

No contexto de colaboração, o valor da centralização de uma rede oferece indicadores relacionados ao seu membro mais central: se ele se relaciona com outros membros centrais e, portanto, parece coordenar toda a rede ou se a rede se encontra mais distribuída e a influência de um ator central não é tão grande (Figura 2.3).

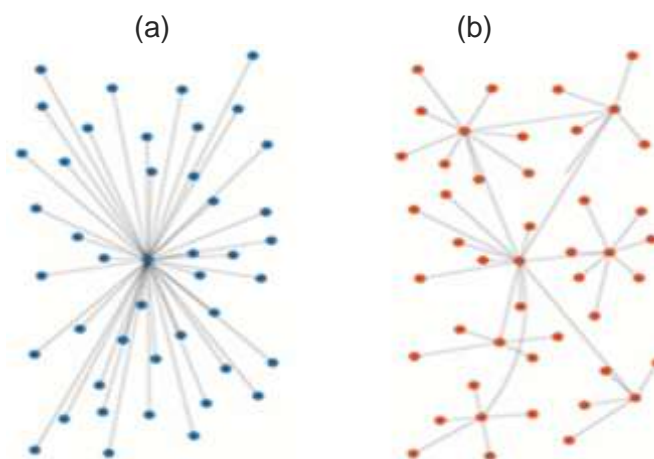


Figura 2.3. Exemplo de uma rede social centralizada (a) e descentralizada (b) Fonte: (ALCANTARA, 2010).

Além dessas medidas avaliadas sobre a rede como um todo, em grupos e organizações sociais é natural que algumas pessoas se destaquem mais e manifestem uma liderança maior sobre outras. As análises de redes sociais também possuem indicadores do prestígio de um membro em relação aos demais, como o seu grau de centralidade, intermediação e proximidade (FREEMAN, 1979) (WASSERMAN e FAUST, 1994).

O **grau de centralidade** (*degree centrality*) de um membro da rede corresponde ao número de arestas que incidem em um vértice, ou simplesmente ao número de vizinhos de um vértice no caso de redes não-direcionadas (Figura 2.4).

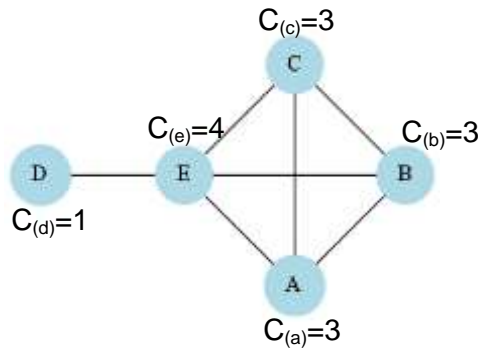


Figura 2.4. Exemplo de rede social com os valores de centralidade de cada ator.

O grau de centralidade no contexto de colaboração pode indicar a importância de um determinado ator da rede. Um ator com alta centralidade concentra um grande número de relacionamentos e, portanto, pode apresentar influência sobre os demais atores.

A **centralidade de intermediação** (*betweenness centrality*) de um membro da rede equivale ao número de vezes que esse membro aparece nos menores caminhos entre os relacionamentos dos outros vértices, e é dado pela seguinte fórmula, onde  $\sigma_{st}$  equivale ao número total de menores caminhos de um membro  $s$  até um  $t$ , e  $\sigma_{st}(v)$  é o número desses caminhos que passam por um nó  $v$ .

$$G_p(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

A centralidade de intermediação indica quais atores que articulam os demais membros da rede. Um ator com alto valor de centralidade de intermediação serve como a única ponte entre grandes grupos de atores e tem um grande poder de influência na coordenação da rede, enquanto atores com baixa centralidade de intermediação são aqueles que se relacionam com grupos que muitos outros membros também se relacionam.

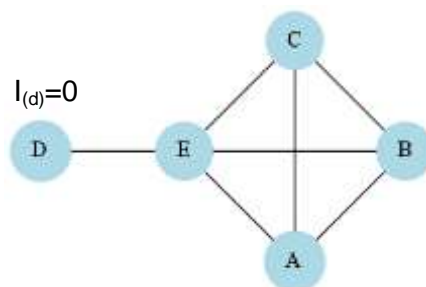


Figura 2.5. Exemplo de rede social, com destaque para o valor de grau de intermediação de um ator equivalente a zero.

Finalmente, a **centralidade de proximidade** (*closeness centrality*) indica a proximidade de um determinado membro na rede em relação aos demais, ou seja, se ele é capaz de interagir diretamente com outro membro. A centralidade de proximidade é dada pela seguinte fórmula, onde  $c(v, t)$  é a distância entre os vértices  $v$  e  $t$ .

$$G_i(v) = \frac{1}{\sum_{t \in V} c(v, t)}$$

A centralidade de proximidade tem uma interpretação próxima ao grau de centralidade, e indica quais atores estão mais próximos dos demais, ou seja, não precisam de outros membros da rede para alcançar um determinado ator. Sendo assim, atores com alto valor de centralidade de proximidade mantêm relacionamento com vários atores e podem exercer muita influência na rede (Figura 2.6), enquanto aqueles com baixo valor precisam relegar suas informações para outros membros.

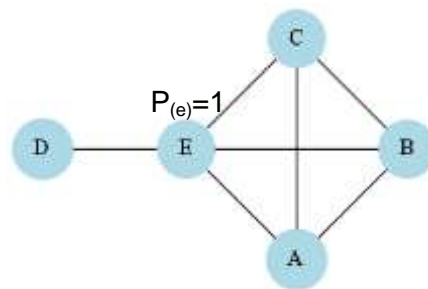


Figura 2.6. Exemplo de rede social, com destaque para o valor de grau de proximidade de um ator.

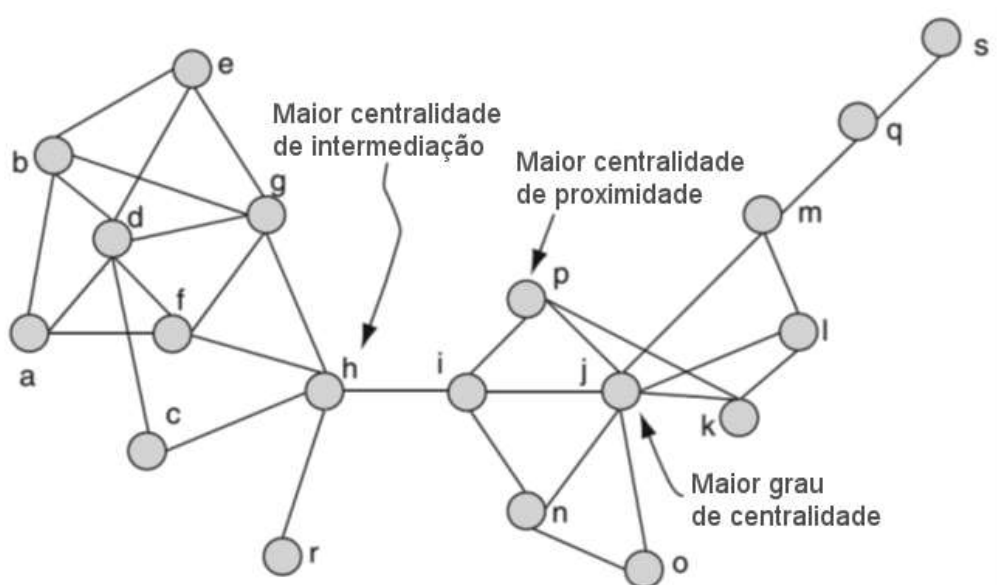


Figura 2.7. Exemplos de diferentes medidas de análise de redes sociais aplicadas (adaptado de (ORTIZ-ARROYO, 2010)).

Como é possível perceber em suas definições e fórmulas, cada uma das medidas apresentadas reflete de uma maneira única alguma característica da rede social, e é comum que estudos as utilizem juntas para ter uma percepção mais completa dos membros e relacionamentos da rede. A **Error! Reference source not found.** ilustra essa diferença, destacando os atores mais importantes segundo diferentes medidas.

## 2.5 Considerações Finais

Esse capítulo discutiu a colaboração enquanto característica crucial para o desenvolvimento de um software. As principais vantagens que podem ser alcançadas através de uma colaboração efetiva são uma maior produtividade no desenvolvimento de software, melhorias na capacidade de resolução de tarefas complexas e aumento da satisfação pessoal dos envolvidos em suas atividades. Entretanto, alcançar essas vantagens não é trivial. Um dos desafios para estimular a colaboração consiste na dificuldade em explicitar como ela ocorre durante a realização de tarefas.

Foi apresentado um panorama de pesquisas relacionadas a colaboração, incluindo como ela pode afetar o desenvolvimento de software, e trabalhos que propuseram técnicas e tecnologias para lidar com essa característica sob o ponto de vista de processos, sendo identificada uma carência de trabalhos relacionados ao acompanhamento da colaboração em processos. Além desses pontos relacionados ao desenvolvimento de software, também foi abordado o uso de propriedades de redes sociais como forma de avaliar algumas características à respeito da colaboração, incluindo uma série de métricas.

O conteúdo desse capítulo é de grande importância para a motivação e fundamentação teórica dessa dissertação, que visa oferecer mecanismos para o acompanhamento e monitoração da colaboração em processos de desenvolvimento de software. A criação desses mecanismos espera favorecer no futuro o planejamento e tomada de decisões em processos considerando aspectos da colaboração. Para alcançar esse objetivo, esse trabalho investiga se técnicas de visualização de software podem ser utilizadas de forma eficiente com essa finalidade, sendo a própria área de visualização de software abordada no próximo capítulo.

## 3 Visualização de Software

*Neste capítulo, a área de Visualização de Software é discutida, com ênfase na visualização de processos de software. São apresentados conceitos importantes e trabalhos encontrados durante a revisão da literatura, que utilizam técnicas de visualização aplicadas à processos.*

### 3.1 Introdução

O desenvolvimento de software é um intrincado universo repleto de problemas frequentemente mal definidos e demasiadamente complexos para a compreensão e construção de soluções (DEGRACE e STAHL, 1998). Acrescenta-se a isso a natureza intangível de sistemas de software que impõe um desafio extra, inibindo a comunicação efetiva de ideias e conceitos entre indivíduos (O'REILLY, BUSTARD e MORROW, 2005). Apesar disso, o desenvolvimento de software é uma atividade humana, e esse universo precisa ser dominado e compreendido por engenheiros para a realização de suas atividades.

A falta de uma imagem coletiva dificulta o estabelecimento de uma base de diálogo comum sobre alguns elementos do software, dificultando a comunicação de detalhes do sistema com *stakeholders* e demais atores da organização. Essa imagem coletiva pode ser proporcionada por visualizações de software, que não têm como objetivo produzir imagens agradáveis, mas sim permitir explorar e memorizar estruturas e características do software observado (DIEHL, 2007).

Nesse sentido, a área de Visualização de Software é o campo de conhecimento que estuda a utilização de técnicas de visualização para exibir artefatos relacionados ao software e seu processo de desenvolvimento (DIEHL, 2007). O uso de representações visuais busca aproveitar a percepção visual e habilidades de identificação de padrão de pessoas para auxiliar no entendimento e desenvolvimento de software, ampliando a cognição dos usuários (CARD, MACKINLAY e SHNEIDERMAN, 1999) (STASKO, 2004).

Técnicas de visualização de software são amplamente utilizadas em várias outras áreas do conhecimento, como química e medicina (DIEHL, 2007). No contexto da Engenharia de Software, estudos empíricos apresentaram diversas evidências de que representações visuais de um produto de software podem aumentar significativamente a compreensibilidade e reduzir o custo de produção e manutenção durante o ciclo de vida de um produto (MILI e STEINER, 2002).

Esse capítulo apresenta uma visão geral sobre a área de visualização de software, e está organizado como segue. A Seção 3.2 discute conceitos importantes da área de Visualização de Software; a Seção 3.3 discorre sobre trabalhos que utilizam técnicas de visualização aplicadas à colaboração para auxiliar no desenvolvimento de software; a Seção 3.4 discorre sobre como visualizações têm sido utilizadas no contexto específico de processos de software; enquanto a Seção 3.5 finaliza o capítulo, apresentando considerações finais.

## 3.2 Percepção e Cognição no Desenvolvimento de Software

Desenvolver sistemas de software é uma tarefa árdua, envolvendo uma série de fases inter-relacionadas ao longo do ciclo de vida de produção. Durante cada uma dessas fases, engenheiros de software precisam lidar e entender elementos complexos distintos. A visualização de software surge nesse cenário como um meio importante para ampliar duas capacidades cerebrais que atuam juntas para o entendimento: percepção e cognição. A percepção se refere a capacidade humana de identificar e reconhecer, enquanto a cognição atua no processo de inferir e processar informações utilizando o próprio conhecimento (TACCA, 2011).

A visualização de software é, portanto, um campo que alia conhecimento da fisiologia humana para criar metáforas que evoquem imagens mentais para auxiliar na memorização e entedimento de estruturas ou funções. Dentre as características de percepção e cognição humana que interferem na concepção de visualizações estão: a utilização de cores, formas e tamanhos, que permitem a diferenciação rápida entre elementos exibidos; simetrias, tamanhos relativos e conectividade para identificação de padrões; e escala de cores para intervalos numéricos contínuos (DIEHL, 2007).

Além de questões relacionadas à fisiologia do olho humano, para conseguirem ampliar a percepção e cognição de seus usuários, as visualizações precisam levar em consideração também os objetivos que desejam alcançar. MALETIC, MARCUS e COLLARD (2002) dividem essas características em cinco dimensões de uma visualização: tarefa, audiência, alvo, representação e meio. Essas dimensões englobam as classificações propostas por taxonomias de visualizações muito referenciadas (PRICE, BAECKER e SMALL, 1993) (ROMAN e COX, 1993), e correspondem ao *porquê*, *quem*, *o que*, *como* e *onde* de uma visualização.

- A **tarefa** diz respeito ao porquê a visualização é necessária, ou seja, qual a(s) tarefa(s) da Engenharia de Software se beneficia(m) pela visualização.

Portanto, testes, manutenção e gestão de software são todos valores válidos para a dimensão tarefa de uma visualização;

- A **audiência** diz respeito a quem utiliza a visualização, ou seja, podem ser desenvolvedores, gestores de processo, estudantes, e qualquer outro papel que se beneficie de uma visualização;
- O **alvo** define o que é visualizado, e artefatos, código-fonte e informações de processo são possíveis exemplos de alvos;
- A **representação** diz respeito a forma com que os dados são exibidos, as estruturas visuais que manifestam a informação, incluindo seus filtros e mecanismos de interação. Exemplos são grafos, gráficos cartesianos e modelos de áreas específicas;
- O **meio** é onde a visualização é observada, e inclui monitores, telas grandes e *display walls*, e ambientes de realidade virtual imersivos.

Essas dimensões precisam ser cuidadosamente estudadas para evitar problemas que comprometam a qualidade e eficácia das visualizações. A escolha da representação é um ponto crucial, e deve ser adaptada às habilidades de percepção do alvo e não o contrário (DIEHL, 2007).

Existe ainda o problema da sobrecarga de informações, que ocorre devido ao grande volume de informações disponíveis que podem ser visualizadas. Essa quantidade grande de dados pode facilmente confundir os usuários e impedir a identificação de qualquer coisa além de uma massa incoerente de dados, mesmo com a utilização de metáforas adequadas para visualizar os dados. Para contornar a sobrecarga de informações, é importante que as representações escolhidas disponibilizem filtros e mecanismos eficientes de tratamento de dados aos usuários da visualização, para que eles possam se concentrar nos dados que realmente importam durante uma determinada observação (CHEN, 2006).

A literatura acadêmica é vasta em trabalhos sobre o tema de visualização de software, com vários estudos que de alguma forma lidaram com esses desafios e aplicaram ou construíram metáforas visuais específicas para ampliar a percepção e cognição durante o desenvolvimento de software. Nessa dissertação, a revisão da literatura foi concentrada em trabalhos que tivessem alguma relação com colaboração ou processos de software, e alguns dos trabalhos encontrados estão descritos nas próximas seções.

### **3.3 Visualizações Aplicadas à Colaboração**

No que diz respeito a colaboração, os trabalhos de visualização encontrados foram divididos em três grupos: trabalhos que atuam ampliando a percepção

(*awareness*) dos interessados no projeto; que estudam a evolução das comunidades que desenvolvem software; e que estudam a comunicação e troca de mensagens entre desenvolvedores.

A percepção no desenvolvimento de software pode ser caracterizada como a consciência de um ator do processo em relação as atividades dos demais, proporcionando contexto para suas próprias atividades (DOURISH e BELLOTI, 1992). Vários trabalhos tentaram ampliar a percepção e cognição de atores para reduzir a dificuldade em acompanhar e situar suas próprias atividades e favorecer o trabalho e a coordenação de equipes, como as ferramentas FASTDash (BIEHL, CZERWINSKI, *et al.*, 2007) e RaisAware (COSTA, FEITOSA e DE SOUZA, 2009).

O FASTDash é um *dashboard* que oferece uma visão de todos os artefatos armazenados em um sistema de controle de versão, e enriquece a percepção dos desenvolvedores exibindo quem está realizando alterações ou trabalhou recentemente em um determinado artefato incluindo informações em nível de métodos. O objetivo da ferramenta é possibilitar que atores possam reconhecer e acompanhar mudanças ocorrendo no desenvolvimento de software em tempo real, e para tanto a tela da ferramenta (Figura 3.1) era projetada no ambiente de trabalho.

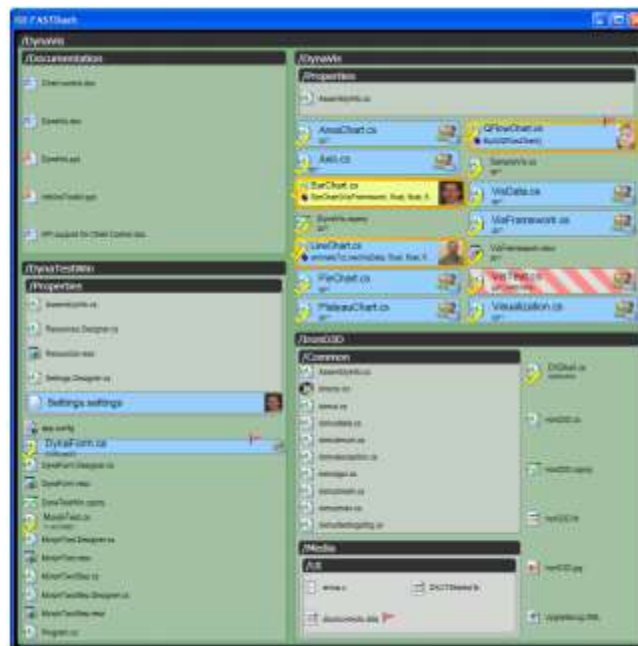


Figura 3.1. Visão da ferramenta FASTDash. Fonte: (BIEHL, CZERWINSKI, *et al.*, 2007).

Outra ferramenta similar é a RaisAware (Figura 3.2), que também coleta dados de repositórios de código fonte e enriquece a percepção dos desenvolvedores sobre o trabalho dos demais. A RaisAware consiste num *plug-in* para a IDE Eclipse que avalia dependências entre artefatos para calcular medidas de análise de redes sociais, e com isso a ferramenta é capaz de indicar ao desenvolvedor arquivos que possuem grande



chance de impactar no trabalho de vários outros desenvolvedores. Além disso, a ferramenta também apresenta grafos que exibem as dependências entre os artefatos, onde são ressaltados potenciais conflitos decorrentes da modificação simultânea de um artefato ou suas dependências por mais de um desenvolvedor.

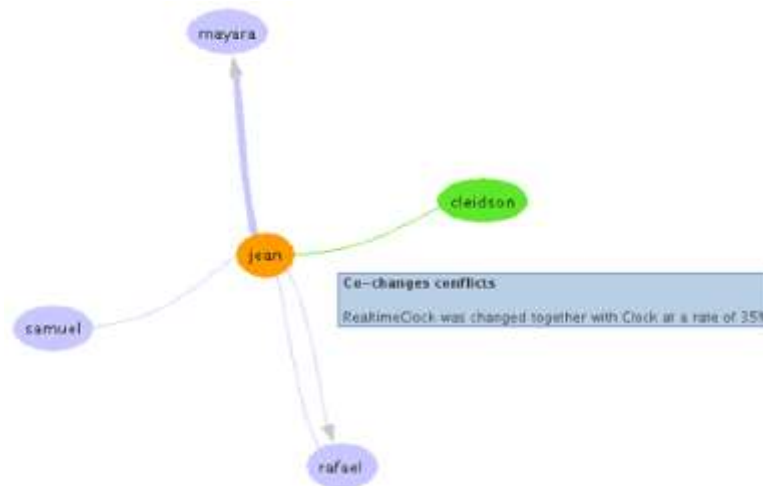


Figura 3.2. Rede social oferecida pela ferramenta RaisAware, com detalhes para potenciais conflitos entre artefatos. Fonte: (COSTA, FEITOSA e DE SOUZA, 2009).

Outros estudos que também visualizam colaboração são aqueles que se dedicam a observar a evolução de sistemas de software. Embora várias ferramentas tratem da evolução de arquitetura e modelos de projetos, sem muita relação imediata com a colaboração, outras focam sobretudo no lado social e na evolução da comunidade que desenvolve o software, estudando os relacionamentos entre os atores envolvidos e a coordenação que o projeto demonstrou ao longo do tempo. São exemplos dessas ferramentas a Augur (DE SOUZA, FROEHLICH e DOURISH, 2005), a EvoTrack-SocialNetwork (VAHIA, MAGDALENO e WERNER, 2011) e a XFlow (SANTANA, OLIVA, *et al.*, 2010).

A ferramenta Augur extrai informações de repositórios de código fonte e possibilita a análise da complexidade de artefatos produzidos durante o projeto. Mais importante para aspectos relacionados à colaboração, a ferramenta sugere relações entre a infraestrutura técnica e social do software, identificando, por exemplo, desenvolvedores que manipulam os artefatos críticos (definidos como o núcleo do projeto) e que por isso possuem potencial para impactar o trabalho de vários outros desenvolvedores. A Figura 3.3 mostra um exemplo de dois atores com comportamentos distintos, um que precisava se coordenar apenas com um outro desenvolvedor e, posteriormente, evoluiu no projeto e passou a ter um papel mais central, enquanto outro exibiu um comportamento inverso e precisava se coordenar com um número menor de outros desenvolvedores.

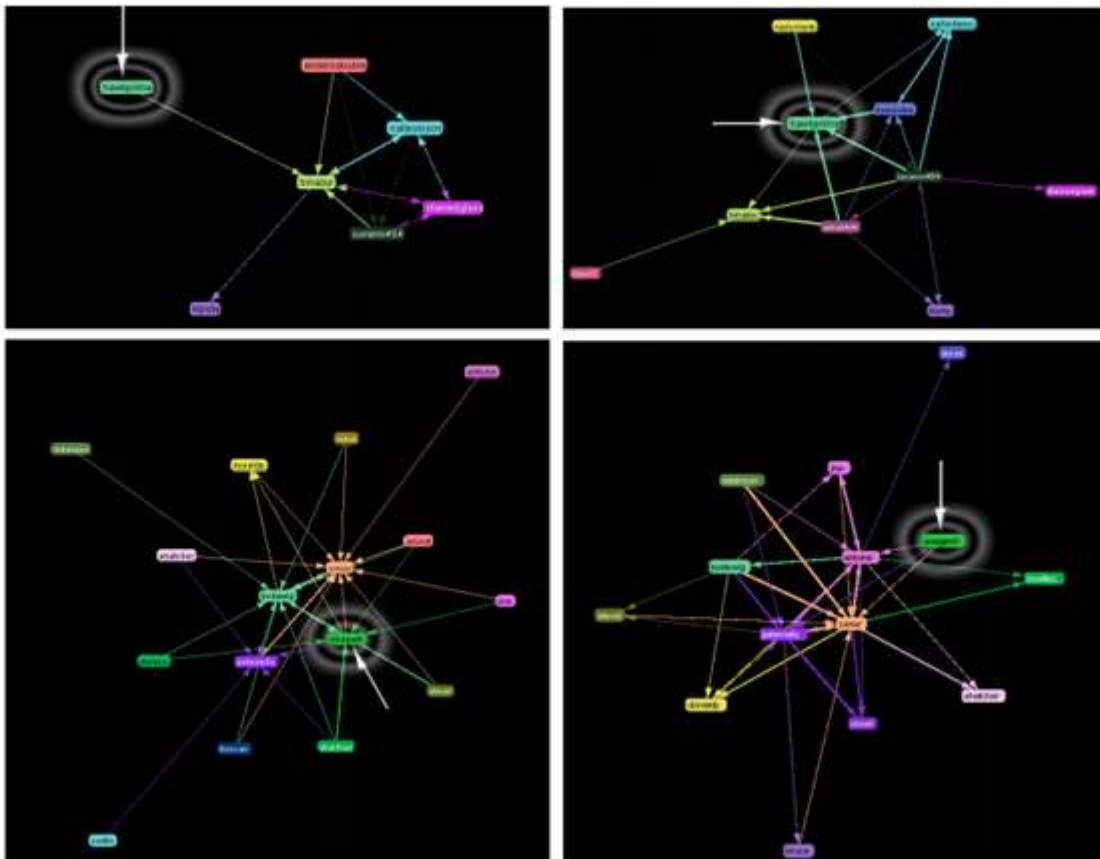


Figura 3.3. Uso de visualizações da ferramenta Augur para estudar mudanças nos relacionamentos entre desenvolvedores de software. Fonte: (DE SOUZA, FROEHLICH e DOURISH, 2005).

A Evoltrack-SocialNetwork é uma ferramenta que está incluída no mesmo contexto dessa dissertação, e compartilha várias similaridades (e.g., ambas são ferramentas desenvolvidas pelo Grupo de Reutilização de Software e são extensões do ambiente EvolTrack, como descrito no próximo capítulo), mas oferecem perspectivas diferentes, com técnicas de coleta e visualização de dados particulares de cada ambiente.

A ferramenta EvolTrack-SocialNetwork é capaz de extrair informações de repositórios de código-fonte e, além do acompanhamento das redes sociais oferecido pela Augur, a EvolTrack-SocialNetwork calcula e apresenta as medidas de redes sociais descritas no Capítulo 2 e oferece vários mecanismos interativos para que o usuário possa interpretar mais facilmente as visualizações, ressaltando nós mais importantes, adições e remoções de uma revisão para outra, entre outras facilidades. A Figura 3.4 exemplifica alguns dos mecanismos que ampliam a percepção do usuário oferecido pela ferramenta, onde o tamanho dos vértices dos usuários é proporcional ao valor de uma métrica (centralidade de grau no exemplo) e cores são utilizadas (a cor azul representa modificação no exemplo).

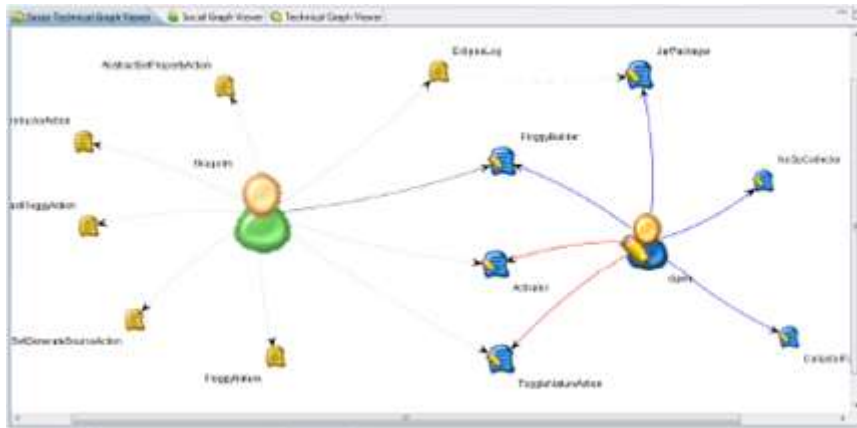


Figura 3.4. Representação de uma rede sociotécnica na ferramenta EvoTrack-SocialNetwork, com recursos de percepção ativados. Fonte: (VAHIA, MAGDALENO e WERNER, 2011).

De forma análoga as duas ferramentas anteriores, a XFlow também coleta informações de repositórios de código-fonte, além de disponibilizar também a extração de dados de repositório de sistemas de gestão de eventos (*Issue Trackers*). Assim como a EvoTrack-SocialNetwork, a XFlow utiliza esses dados para exibir as várias medidas de redes sociais apresentadas no Capítulo 2, oferecendo ao usuário um conjunto de visualizações e mecanismos de interação para facilitar a análise da evolução de projetos de software. A Figura 3.5. mostra um exemplo de três das cinco visualizações oferecidas pela XFlow, exibindo dados de projetos de software livre.

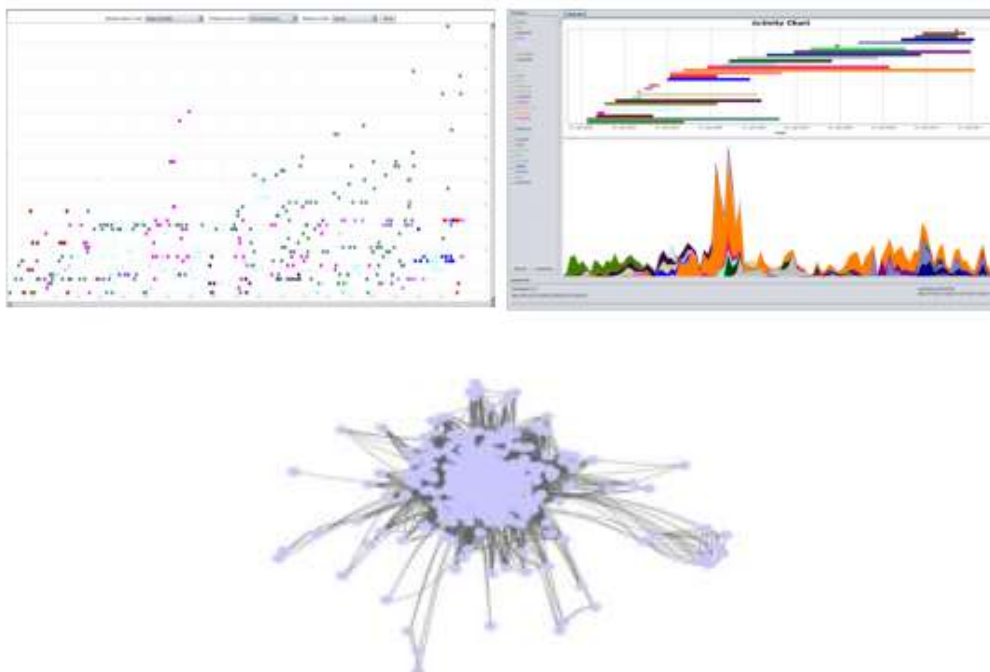


Figura 3.5. Três visualizações exibidas pela ferramenta XFlow, exibindo os relacionamentos dos atores e métricas calculadas sobre a rede social e as dependências entre artefatos. Fonte: (SANTANA, OLIVA, *et al.*, 2010).

Finalmente, existem as ferramentas que funcionam de forma similar as de evolução de software, mas que possuem como alvo de suas visualizações listas de emails e troca de mensagens entre os interessados de um projeto de software. Essas ferramentas são capazes de exibir informações sobre quem se comunicou com quem durante o desenvolvimento de um determinado software e identificar padrões de comunicação dentre as mensagens trocadas. Exemplos dessas ferramentas são a OSSNetwork (BALIEIRO, SOUSA JÚNIOR e DE SOUZA, 2008) e Soylent (FISHER, 2004).

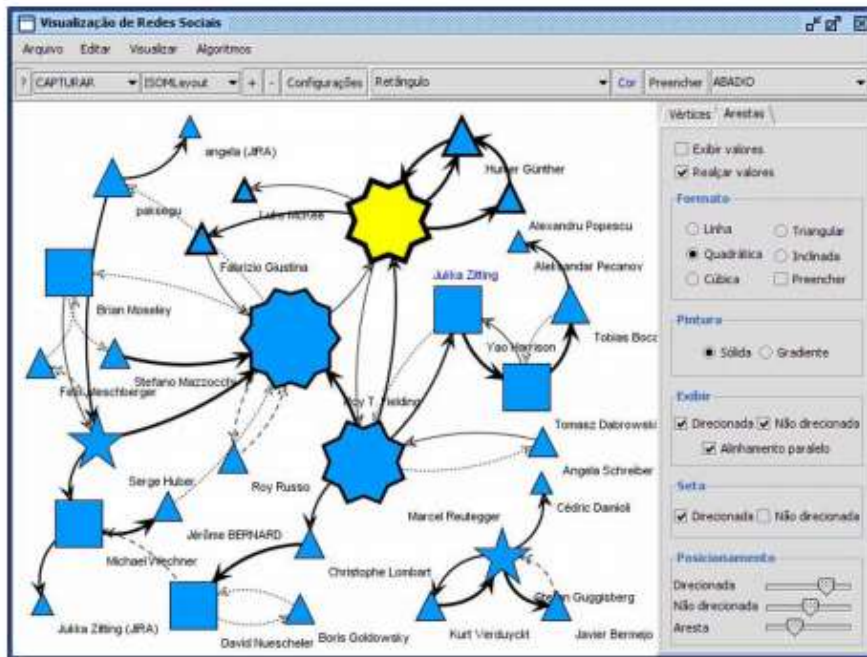


Figura 3.6. Redes sociais de comunicação exibidas pela ferramenta OSSNetwork. Fonte: (BALIEIRO, SOUSA JÚNIOR e DE SOUZA, 2008).

A ferramenta OSSNetwork utiliza *webcrawlers*<sup>7</sup> para recuperar informações em listas de emails e fóruns de discussão de software livre, gerar redes sociais a partir desses dados e exibir as redes em um ambiente que oferece mecanismos para modificar a visualização segundo alguma característica da rede social. A Figura 3.6 mostra um exemplo, com a forma geométrica dos nós sendo modificada de acordo com alguma medida de análise de rede social diferente.

A ferramenta Soylent também revela a colaboração entre atores que participaram de um desenvolvimento de software através da troca de mensagens entre

<sup>7</sup> *Webcrawlers* são pequenos programas criados com o objetivo de recuperar automaticamente informações de páginas web, conseguindo navegar por várias páginas de acordo com o padrão codificado (PINKERTON, 2000).

eles. Utilizando a ferramenta, pesquisadores foram capazes de identificar várias estruturas de comunicação (

Figura 3.7), incluindo projetos onde: i) existem dois grupos bem distintos, um que monopoliza grande parte da discussão sobre o projeto e outro que recebe poucas mensagens e contata poucos outros membros fora do primeiro grupo; ii) existem alguns poucos indivíduos que centralizam e agem como pontes entre os demais; iii) existem dois grupos bem definidos que não trocam mensagens entre si, e são intermediados por um único ou um pequeno grupo de outros membros da rede.

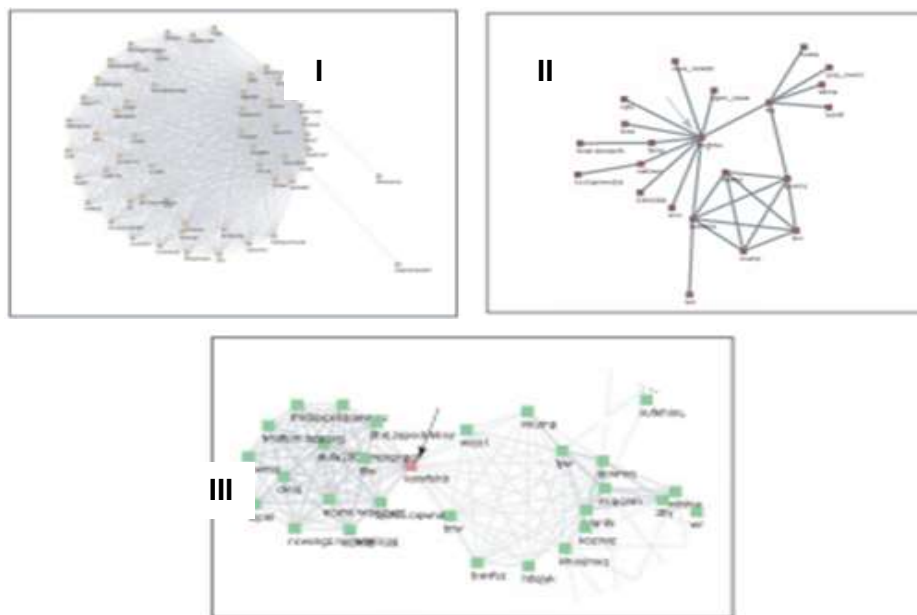


Figura 3.7. Padrões de comunicação encontrados por meio da ferramenta Soylent. Fonte: (FISHER e DOURISH, 2003)

### 3.4 Visualização no Contexto de Processos

Embora grande parte dos trabalhos relacionados apresentem visualizações tendo tarefas relacionadas ao desenvolvimento, um pequeno grupo foi encontrado que tinha o objetivo de enriquecer a percepção de usuários acerca de processos de software ou ainda processos de negócio.

Panas, Berrigan e Grundy (2003) utilizaram a metáfora de cidades virtuais (WETTEL e LANZA, 2008) (Figura 3.8) para ilustrar a produção de um software. Embora não utilizem dados de processo, os autores afirmam que as informações retratadas são de interesse tanto de desenvolvedores quanto de gerentes, exibindo dados de artefatos de código-fonte para ilustrar o andamento do desenvolvimento. Na visualização, o tamanho dos prédios reflete o tamanho de um determinado artefato, enquanto incêndios e relâmpagos indicam componentes muito executados e muitas vezes modificados. O trabalho é apresentado como uma pesquisa em andamento e,

em seus trabalhos futuros, são listadas possibilidades mais relacionadas aos processos, como informações relacionadas ao custo e esforço de atividades, mas nenhum trabalho posterior sobre o assunto foi encontrado.



Figura 3.8. Cidades virtuais em três dimensões utilizadas para retratar a produção de um software (PANAS, BERRIGAN e GRUNDY, 2003).

Rilling *et al.* (2007) apresentam um ambiente para visualização de processos de manutenção de software. O objetivo do ambiente é estabelecer uma rastreabilidade entre atividades de um processo para os recursos disponíveis (ferramentas, técnicas e atores). A Figura 3.9 mostra o ambiente proposto, exibindo: as seis fases do processo padrão de manutenção definido pela IEEE (IEEE, 2006) Figura 3.9(A); uma visão detalhada das atividades e subatividades de manutenção Figura 3.9(B); as ferramentas aplicáveis que estão disponíveis para lidar com determinada tarefa (coloridas de branco), com o cone representando o contexto do processo (que consiste em uma lista predefinida de situações responsáveis por reduzir a quantidade aplicável de soluções) aparecem na Figura 3.9(C); e informações de dependências estruturais seguindo uma ontologia Figura 3.9(D).

Suntinger *et al.* (2007) apresentam o *Event Tunnel*, uma visualização em três dimensões criada para que analistas possam visualizar eventos que ocorreram em um processo e responder questões tais quais onde/quando ocorreram anomalias no processo, qual caminho de execução parece ser mais eficiente, em quais pontos o processo parece ficar mais lento, entre outras. A abordagem não é voltada especificamente para sistemas de software, tendo sido avaliada no contexto de processos de logística e de detecção de fraudes. O ambiente coleta dados de repositórios com os eventos reportados e os exibe com diferentes recursos para que o usuário possa analisá-los. A **Error! Reference source not found.** mostra a ferramenta em funcionamento, exibindo todos os eventos em uma visão vertical (**Error! Reference source not found.** (a)) e de forma horizontal (**Error! Reference**

source not found. (b)); o texto de um determinado evento (**Error! Reference source not found. (c)**) e várias outras visões para que o usuário possa compreender melhor as relações e efeitos de um determinado acontecimento em um processo.

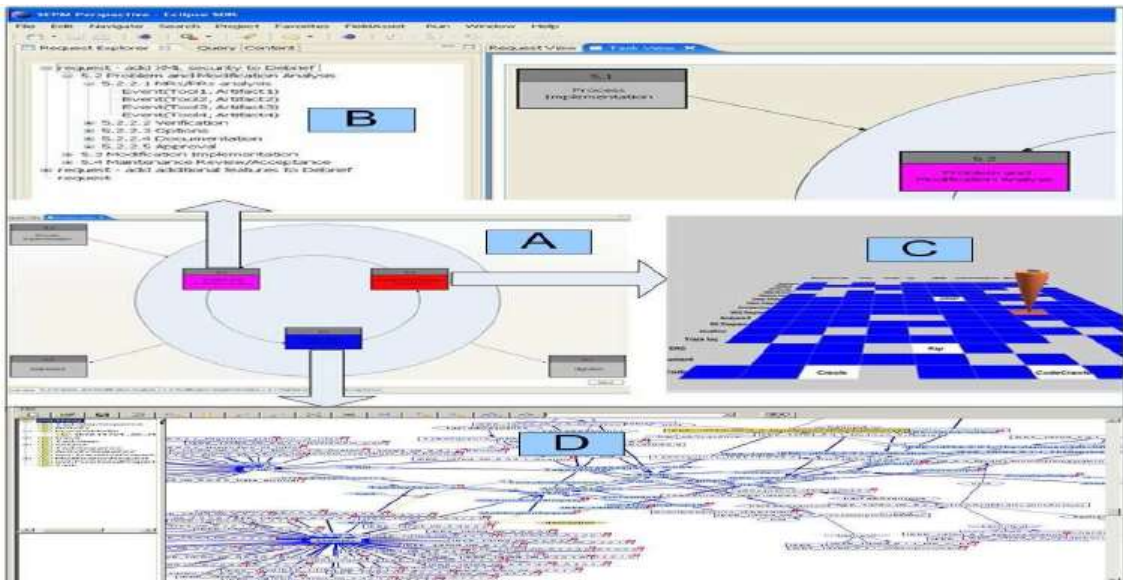


Figura 3.9. Múltiplas visões oferecidas em (RILLING, MENG, *et al.*, 2007) para rastrear solicitações de mudanças em um processo de manutenção.

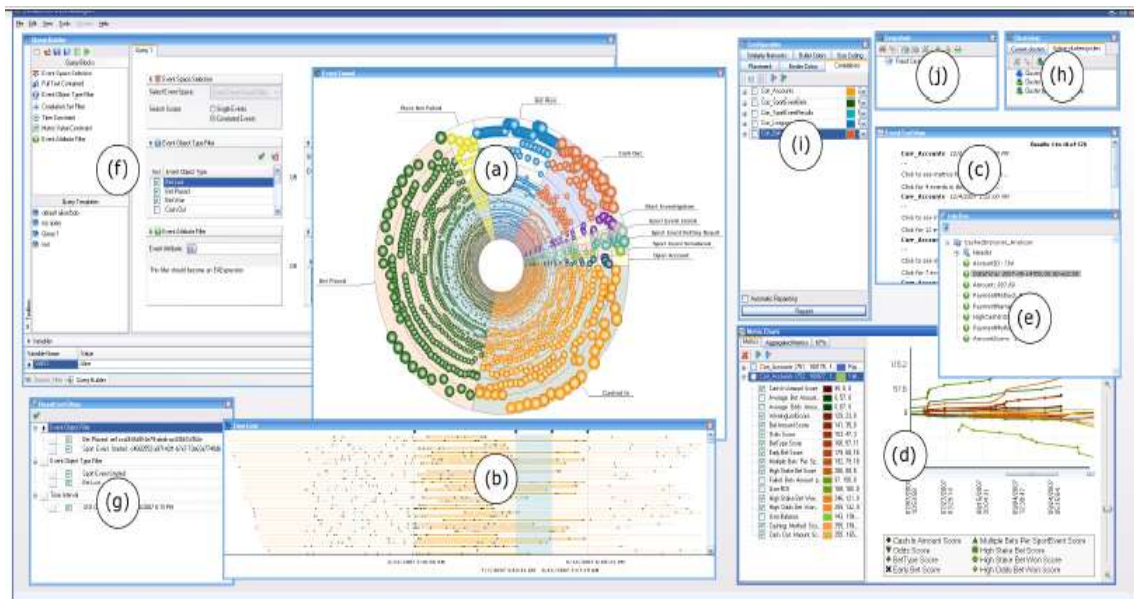


Figura 3.10. Eventos exibidos na ferramenta *Event Tunnel*.

Brown, Recker e West (2011) apresentam uma abordagem inovadora para modelagem de processos, utilizando um ambiente virtual 3D construído sobre o

mundo virtual online SecondLife<sup>8</sup>. A abordagem oferece uma série de recursos para que usuários possam modelar processos de forma colaborativa dentro do ambiente virtual, utilizando recursos visuais bem diferentes das outras abordagens (Figura 3.11).



Figura 3.11. Modelagem colaborativa de processos através de um ambiente de realidade virtual.

Há também um conjunto de trabalhos destinados à visualização de *workflows*, tema muito similar aos processos. *Workflows* são modelos empregados em diversas áreas de conhecimento para auxiliar na explicitação e automatização de eventos que se repetem, definem uma sequência de tarefas para alcançar um objetivo (DEELMAN, GANNON, *et al.*, 2009).

Nessa linha de trabalho, as técnicas de visualizações são utilizadas com propósito de ressaltar as atividades de um *workflow* ou reconstruí-lo a partir de logs de execução. (BOBRIK, REICHERT e BAUER, 2007) é um exemplo (Figura 3.12), utilizando grafos para a representação de *workflows*. A ferramenta também é interativa, permitindo que os usuários possam definir predicados para, por exemplo, destacar as atividades que um determinado ator não esteve envolvido.

---

<sup>8</sup> SecondLife é um ambiente virtual online gratuito. Mais informações em: <http://secondlife.com>



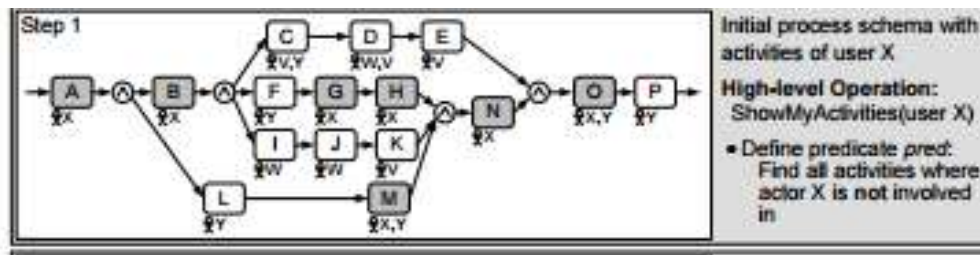


Figura 3.12. Visualização de um *workflow* ressaltando apenas atividades em que um determinado ator não esteve envolvido. Fonte: (BOBRIK, REICHERT e BAUER, 2007).

Existe também a ferramenta Ambrose<sup>9</sup> (Figura 3.13), lançada como software livre e registrada pelo grupo Twitter. A Ambrose foi desenvolvida para exibir *workflows* via *web*, sendo capaz de mostrar uma barra de progresso da completude do *workflow* inteiro e dependências entre as atividades, além de disponibilizar mecanismos de percepção de modo similar à ferramenta EvolTrack-SocialNetwork, tornando vértices maiores com base na quantidade de recursos que consomem.

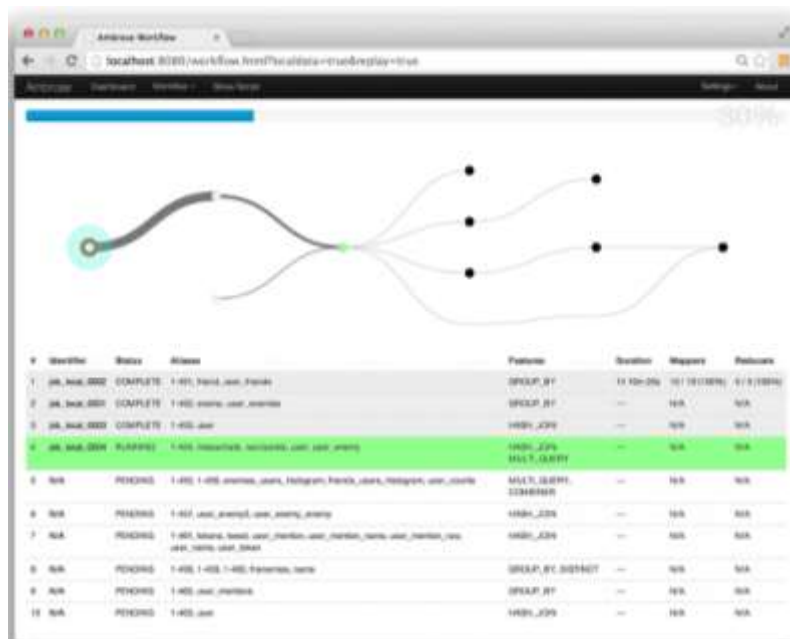


Figura 3.13. Ferramenta Ambrose exibindo um *workflow*. Fonte: <https://github.com/twitter/ambrose>

Embora esses trabalhos de visualização de *workflows* tenham certa similaridade com a proposta dessa dissertação, representando atividades e fluxos para a produção de algum produto ou resultado, o foco das abordagens é bem diferente. Pelo fato de ser voltado para a execução automatizada, os trabalhos em *workflows*

<sup>9</sup> <https://github.com/twitter/ambrose>

não dão muita importância ao trabalho colaborativo, crucial para o desenvolvimento de software e principal foco dessa dissertação.

### 3.5 Considerações Finais

Esse capítulo apresentou a área de Visualização de Software, um tema de grande importância para essa dissertação já que seu objetivo é a construção de um ambiente que ofereça visualizações para o entendimento e acompanhamento da colaboração em processos de software. Técnicas de visualização já são aplicadas em várias outras áreas de conhecimento, com diversos estudos empíricos que indicam o potencial que as técnicas oferecem para aumentar a compreensibilidade, identificação de padrões e redução de sobrecarga cognitiva ao processar informações.

Foram descritos os vários trabalhos encontrados na revisão da literatura que buscou por técnicas de visualização aplicadas para lidar com a colaboração no desenvolvimento de software, com ênfase em particular nos trabalhos que apresentavam uma visão dos processos de desenvolvimento.

A Tabela 3.1 apresenta um comparativo entre os trabalhos de visualização de processos de software, ressaltando seus focos diferentes. No entanto, é possível notar um fato comum a todas essas visualizações que de alguma forma são aplicadas a processos: em todas elas, o foco das visualizações sempre é para responder tarefas relacionadas ao estado de um processo (e.g., Está atrasado? Quantas atividades falharam? Onde falhou?) e não sobre a equipe e colaboração da organização (e.g., quem está executando aquela atividade? Com quem? Onde estão localizados? Quais instâncias eles atuaram e em quais atividades?). O fato de não ter sido encontrada nenhuma ferramenta que fosse voltada para tarefas colaborativas contribuiu significativamente para a motivação dessa dissertação.

O ambiente EvolTrack-Process é o foco do próximo capítulo, tendo sido construído sobre a fundamentação teórica de visualização de software e colaboração apresentada nesses dois últimos capítulos.

Tabela 3.1 Comparativo de abordagens de visualização de processos.

Referência	Tarefa	Audiência	Alvo	Representação	Meio
(PANAS, BERRIGAN e GRUNDY, 2003)	Manutenção de Software	Desenvolvedores, Arquitetos de Software <sup>†</sup>	Artefatos de código-fonte	Cidades Virtuais	Monitores
(RILLING, MENG, <i>et al.</i> , 2007)	Manutenção de Software	Gerentes de Processo	Processos de Manutenção	Grafo, metáforas próprias	Monitores

(SUNTINGER, OBWEGER, <i>et al.</i> , 2007)	Compreensão de Eventos de Processos	Gerentes de Processo	Eventos de Processo	<i>Event Tunnel</i> , gráfico de linha	Monitores
(BROWN, RECKER e WEST, 2011)	Modelagem de Processos	Gerentes de Processo	Elementos de Modelagem de Processos	Modelo de Processo em Ambiente de RV	Monitores

† Os autores afirmam que a abordagem também tem foco para tarefas e papéis associados à processos de software.

## 4 EvoTrack-Process

*Este capítulo detalha o ambiente proposto nessa dissertação, denominado EvoTrack-Process. São apresentados o contexto do trabalho e detalhes de como a colaboração pode ser extraída e acompanhada por meio de visualizações de processos, bem como de que forma essas informações podem ser trabalhadas para facilitar a interpretação da colaboração.*

### 4.1 Introdução

Facilitar a colaboração entre os times de uma organização não é uma tarefa trivial na área da Engenharia de Software. A partir das revisões da literatura discutidas nos capítulos anteriores, foram encontrados diversos trabalhos que destacaram alguns desafios que surgem naturalmente durante o desenvolvimento de software e podem dificultar a ocorrência da colaboração efetiva entre os atores sem adoção de nenhum mecanismo de apoio (DEFRANCO-TOMMARELLO e DEEK, 2002) (WHITEHEAD, 2007).

Dentre os mecanismos encontrados nas revisões da literatura para lidar com esse problema, foi identificado que as soluções são limitadas em relação ao suporte que oferecem para o entendimento da colaboração de processos. Em geral, as soluções identificadas sequer coletam informações de *logs* ou repositórios com informações específicas de processos, e, portanto, não disponibilizam esses dados para entender e acompanhar como a colaboração ocorreu durante a execução de processos (DE SOUZA, FROEHLICH e DOURISH, 2005)(BIEHL, CZERWINSKI, *et al.*, 2007) (COSTA, FEITOSA e DE SOUZA, 2009) (VAHIA, MAGDALENO e WERNER, 2011). Ignorar essa informação pode resultar numa visão incompleta da colaboração, podendo ser usada como base para intervenções que acabem prejudicando ou acrescentando pouco ao desenvolvimento do software, como a identificação de atores que aparentam ser pouco relevantes ou que estejam ociosos, quando na verdade eles não possuem atividades planejadas no fluxo que a instância do processo seguiu.

Assim, surgiu o ambiente proposto nessa dissertação, EvoTrack-Process (SANTANA, 2014). O EvoTrack-Process foi projetado para auxiliar a compreensão e acompanhamento da colaboração em processos de software, tornando a colaboração mais efetiva dentro da organização através do uso de técnicas de análise de redes sociais e de visualização de processos.

Para apresentação do ambiente, o restante do capítulo está organizado como segue: a Seção 4.2. descreve o contexto de pesquisa no qual essa dissertação foi concebida; a Seção 4.3. apresenta uma visão geral dos principais elementos que compõem o ambiente construído; a Seção 4.4. é dedicada a descrever os dados coletados pelo ambiente, apresentando de que forma a colaboração pode ser extraída; a Seção 4.5 comenta as medidas de colaboração de processos utilizadas; e, por fim, a Seção 4.6. trata da interpretação e visualização da colaboração em processos.

## 4.2 Contexto

Anteriormente a esse trabalho, a área de processos de software foi diretamente explorada no Projeto CDSOft<sup>10</sup> (WERNER, ARAUJO e SANTORO, 2011), um projeto de pesquisa que envolveu 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 CDSOft foi direcionado para conceber uma sistemática e o devido apoio computacional para o balanceamento de colaboração e disciplina em processos de software. As soluções propostas abrangem as áreas de engenharia de linhas de processos, gestão de contexto e gestão da colaboração em processos de software (Figura 4.1).

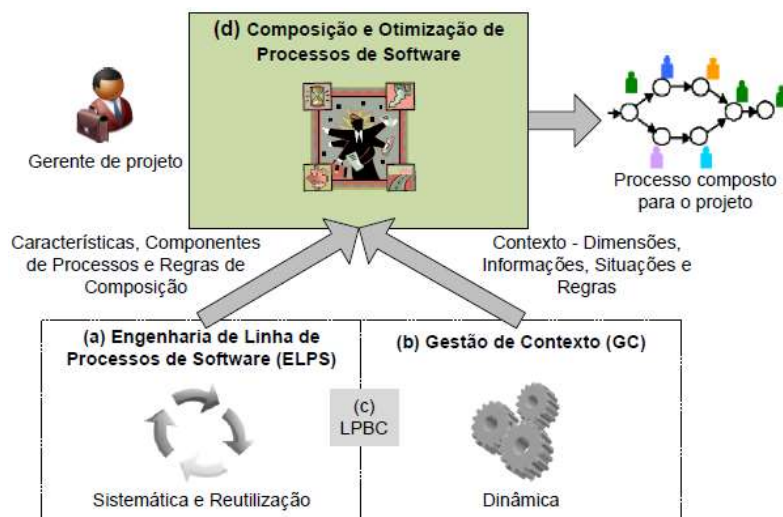


Figura 4.1. Visão geral das soluções propostas pelo Projeto CDSOft

<sup>10</sup> Mais informações disponíveis em: <http://reuse.cos.ufrj.br/cdssoft/>

O ambiente EvolTrack-Process foi planejado para expandir especificamente uma das propostas do Projeto CDSOft, que tinha, entre seus objetivos, a gestão da colaboração em processos de software: a COMPOOTIM.

#### 4.2.1 COMPOOTIM

A COMPOOTIM (MAGDALENO, 2013) é uma sistemática de gestão da colaboração baseada em quatro etapas: planejamento, composição, otimização e acompanhamento da colaboração, conforme apresentado na Figura 4.2. Cada uma dessas etapas foi criada para resolver algum desafio da introdução da colaboração nos processos de software, e, para cada uma delas, foram desenvolvidas ou aplicadas soluções para alcançar os objetivos da etapa.

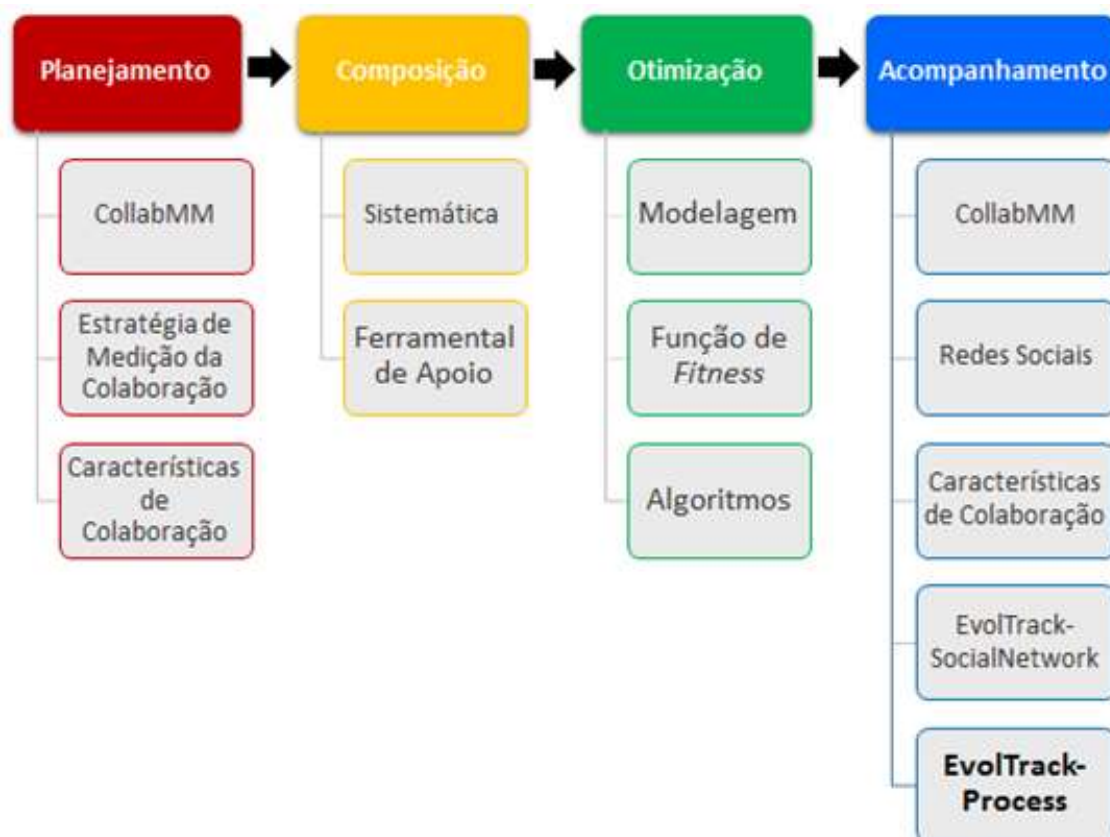


Figura 4.2. Visão geral da COMPOOTIM incluindo o ambiente Evoltrack-Process.

A etapa de **planejamento** da colaboração permite às organizações de software estabelecer objetivos e definir ações para atingir níveis mais altos de colaboração. São estabelecidas algumas atividades de planejamento nas quais o gerente de projetos define o nível de colaboração desejado de cada projeto e realiza uma composição inicial do processo. 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. Durante a execução, é necessário

também avaliar se o nível planejado de colaboração ainda se mantém como definido e, se for o caso, rever o planejamento feito anteriormente.

A etapa de **composição** de processos deve ser capaz de sugerir um processo específico para o projeto, a partir de componentes de processo, de acordo com o contexto do seu desenvolvimento.

A etapa de **otimização** tenta 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, propondo ao gerente de projeto um processo que satisfaça as restrições do contexto do projeto.

O **acompanhamento** da colaboração diz respeito a monitoração do nível real de colaboração em um determinado processo, verificando se a colaboração está de fato acontecendo conforme o planejado e avaliando os resultados alcançados com a colaboração no processo. Durante essa etapa, o gerente de projeto é munido de informações sobre a execução do projeto para tomar decisões e possíveis ações corretivas.

O ambiente EvolTrack-Process surge para preencher algumas lacunas referentes a colaboração em processos que não foram abordadas pela COMPOOTIM na etapa de acompanhamento, conforme ilustrado na Figura 4.2. Em especial, algumas funcionalidades relacionadas à monitoração da colaboração oferecidas pela COMPOOTIM não possuem integração direta com elementos de processos, inferindo a colaboração apenas a partir de artefatos de código fonte através da EvolTrack-SocialNetwork (VAHIA, MAGDALENO e WERNER, 2011). Sendo assim, o ambiente EvolTrack-Process permite que a fase de acompanhamento da COMPOOTIM tenha uma visão mais ampla do desenvolvimento de software, considerando aspectos técnicos e sociotécnicos tanto sob a perspectiva dos artefatos produzidos quanto das instâncias de processo e suas atividades.

### 4.3 Visão Geral

O EvolTrack-Process (SANTANA, 2014) consiste em um ambiente para acompanhamento e monitoração da colaboração. O ambiente amplia os mecanismos já propostos pela COMPOOTIM, atuando mais especificamente na fase de acompanhamento e expandindo a integração da sistemática com ambientes de processo.

O funcionamento geral do Evoltrack-Process segue três etapas principais: a (i) extração de dados; (ii) medição de colaboração através de métricas de análise de redes sociais; (iii) apresentação de visualizações de processos com foco em

colaboração (Figura 4.3). O ambiente também possibilita ao usuário o controle sobre qual instância de processo será observada, quer essa instância já tenha sido finalizada ou ainda esteja em andamento.

A respeito da etapa de **extração de dados**, o ambiente prevê mecanismos de coleta automática de dados contidos em repositórios de processo. A partir desses dados, são extraídos os modelos de processo e informações (atividades, atores e artefatos) de execução da instância, complementando outra solução existente que apoia a sistemática COMPOOTIM.

Em relação à **medição da colaboração**, o EvoITrack-Process utiliza as propriedades de análise de redes sociais (WASSERMAN e FAUST, 1994), descritas no Capítulo 2, para inferir a importância do relacionamento entre os atores em uma determinada instância do processo, assim como medidas gerais da rede que consideram todos os relacionamentos e dão uma visão absoluta da colaboração ocorrida na instância.

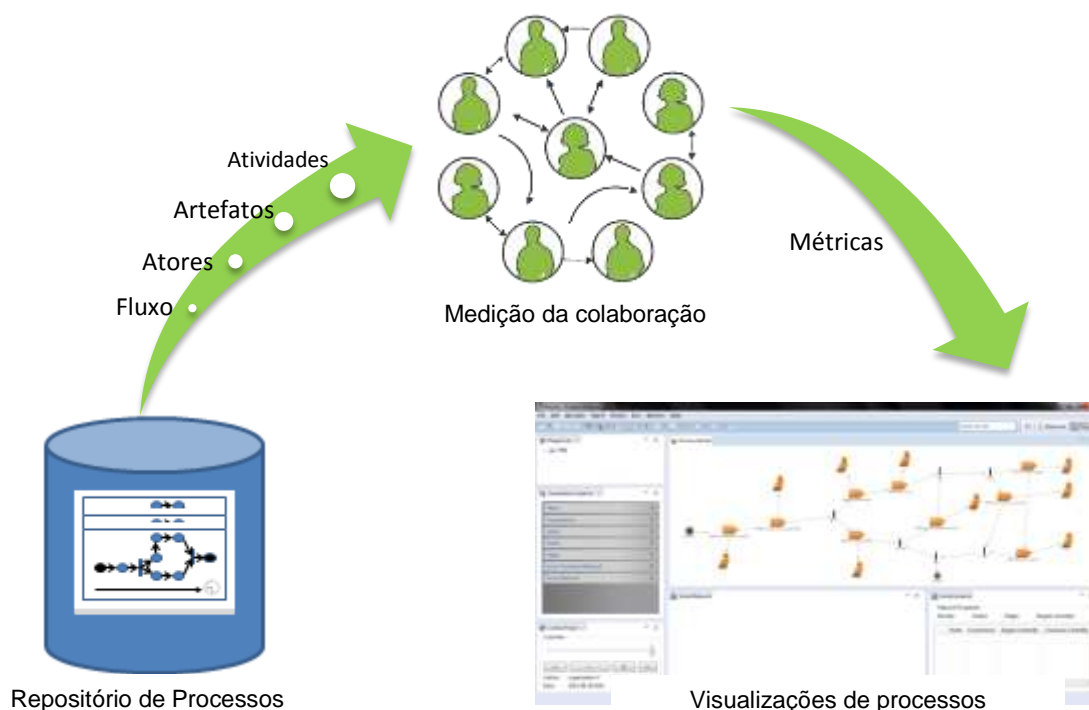


Figura 4.3. Visão geral do ambiente EvoITrack-Process.

Por fim, a etapa de **visualização de processos** é responsável por apresentar os dados e ressaltar a colaboração de forma visual. A partir da reconstrução do modelo do processo seguindo os *logs* de como ocorreu a execução, as visualizações têm o objetivo de identificar quais atores efetivamente trabalharam para a realização das atividades seguindo o fluxo adotado. Além disso, é construída também uma rede social representando os atores que trabalharam em equipe para a realização de uma



mesma atividade com algumas informações adicionais, como por exemplo a distribuição geográfica dos atores.

Nas próximas seções desse capítulo cada uma das etapas do ambiente será descrita com mais detalhes, discutindo as técnicas que foram utilizadas para alcançar seus objetivos e suas vantagens, desvantagens e limitações.

## 4.4 Extração de Dados

Embora dados coletados indiretamente tenham a desvantagem de serem normalmente incompletos, retratando *o que* aconteceu e não *os motivos* por trás das decisões tomadas, a utilização de dados de repositórios tem se mostrado viável para revelar características de como ocorreu o desenvolvimento de modo não intrusivo ao processo (BALL, KIM, *et al.*, 1997). Essas técnicas diminuem o revés da obtenção de dados junto aos envolvidos (KOCH, 2007), que podem, por exemplo, não recordar precisamente como o desenvolvimento ocorreu ou não ter uma visão ampla de como o processo foi conduzido, além da vantagem de facilidade ao acesso desses dados de repositórios, que permitem que um grande volume de informação seja extraído.

Tendo em mente essas vantagens e desvantagens das formas direta e indireta de coletas de dados, optou-se pelo uso da coleta indireta para o EvoITrack-Process. Dessa forma, a aquisição de dados se torna mais fácil, sobretudo diante de ocasiões onde o ambiente tem sua utilidade ampliada, como no caso de software desenvolvido de forma distribuída e o contato com os indivíduos envolvidos é mais complexo. A coleta de dados a partir de repositórios também favorece a observação de instâncias passadas uma vez que não depende da memória dos envolvidos, outra característica importante do ambiente. Essas vantagens foram determinantes para o projeto do ambiente, resultando no estabelecimento do primeiro requisito funcional do EvoITrack-Process: a coleta de dados de processos automatizada a partir de repositórios ambientes de execução de processos.

Ainda relacionado ao desafio da extração de dados, a necessidade de lidar com um grande volume de informações deu origem a outro requisito funcional do ambiente: a coleta de dados incremental. A grande quantidade de dados que pode ser extraída de um repositório com informações de desenvolvimento de software exige um ferramental de apoio que auxilie o usuário nessa tarefa, indo além da coleta e processamento de dados brutos (BEVAN, WHITEHEAD, *et al.*, 2005). Em particular, se faz necessário que os trabalhos possibilitem que o usuário acrescente novas informações ao corpo de dados já coletado, sendo capazes de agregar informações novas e, dessa forma, tornando possível análises mais completas à medida em que novos dados se tornam disponíveis.

Nesse cenário, os trabalhos de mineração de processos (VAN DER AALST, REIJERS, *et al.*, 2007) têm como objetivo a identificação automática de relações a partir de *logs* de execução de processos. Embora grande parte dessas ferramentas utilize algumas técnicas que também são importantes para essa dissertação, sobretudo relacionadas à coleta de dados de processos, os objetivos das ferramentas são diferentes: enquanto ferramentas de mineração visam extrair relações relevantes automaticamente, nossa proposta tem como intenção oferecer visualizações para que o usuário tire suas próprias conclusões com base em sua experiência, além de poder ser utilizada sem a necessidade de um grande volume de dados de execuções de processos anteriores.

Uma vez definido que a fonte de dados do ambiente são repositórios de execução de processos, a unidade básica de informação passa a ser o *log* de execução das instâncias de processo. Embora não exista um modelo único desses *logs*, já que cada ferramenta de processo tem seus próprios critérios sobre quais informações são relevantes, é comum achar uma estrutura básica de elementos de processo sob três perspectivas distintas (VAN DONGEN, DE MEDEIROS, *et al.*, 2005): a perspectiva de processo, com informação de controle e ordem de execução das atividades, possibilitando a identificação de todos os fluxos possíveis e a reconstrução do modelo de processo executado; a perspectiva organizacional, que identifica quem realizou cada uma das atividades, e indica qual papel cada um dos atores assumiu; e a perspectiva do caso, que identifica a instância executada e permite a associação correta dos dados de cada uma das execuções.

A partir de informações básicas disponibilizadas em *logs* da maioria das ferramentas, outras características podem ser derivadas dependendo da riqueza de informação desses *logs*, como por exemplo o tempo de execução da instância e o desvio de estimativa de prazo. Sob o ponto de vista organizacional, informações extras sobre a localidade dos atores são suportadas por algumas ferramentas e podem ser especialmente úteis para compreender melhor características relacionadas à colaboração, influenciando diretamente nos requisitos mínimos que as fontes de dados escolhidas deveriam atender.

A Figura 4.4 apresenta o modelo de dados projetado para representar um processo coletado. Nele, os elementos de processo são atividades, artefatos, atores e elementos de fluxo, e cada um desses elementos pode estar ligado a vários outros elementos de processo, estabelecendo o fluxo geral do modelo. Essas entidades são as mesmas adotadas por outras abordagens de mineração de processos (VAN DONGEN, DE MEDEIROS, *et al.*, 2005) (RUBIN, GÜNTHER, *et al.*, 2007), com

exceção dos elementos de fluxo, estes últimos adicionados por serem importantes para a construção de uma das visualizações.

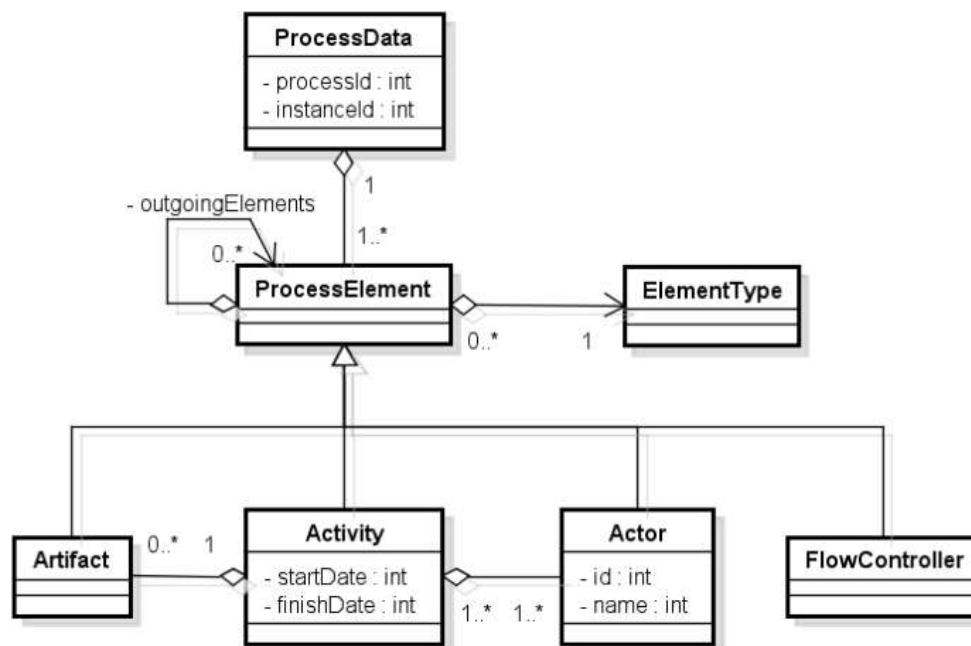


Figura 4.4. Modelo de dados do ambiente EvoITrack-Process.

Como nem todas as ferramentas de automação de processos oferecem o mesmo nível de detalhes em seus *logs*, as entidades do modelo foram criadas com o propósito de serem flexíveis, ou seja, gerais o suficiente para permitir o funcionamento com um conjunto mínimo de informações para garantir a compatibilidade com o maior número de ferramentas, enquanto ao mesmo tempo permitindo a manipulação de alguns elementos específicos, como os dados da localidade dos atores discutidos anteriormente, que são opcionais para a utilização do EvoITrack-Process e tornam disponíveis algumas funcionalidades, caso estejam presentes. Essa tentativa de garantir a compatibilidade com o maior número de ferramentas de automação de processos e seus repositórios foi uma preocupação para que o ambiente EvoITrack-Process pudesse se manter útil ao longo de seu ciclo de vida, resultando na definição de mais um requisito: o suporte para dados mantidos em diferentes repositórios.

A ferramenta escolhida para desenvolvimento do primeiro conector de dados do ambiente foi o BPMS Bonita. Desenvolvida para lidar com processos de negócio de diversas naturezas, a notação BPMN e, conseqüentemente, ferramentas de BPMS's, também tem sido muito utilizadas no cenário do desenvolvimento de software (RECKER, 2010). O BPMS Bonita<sup>11</sup> é uma dessas ferramentas lançada como

<sup>11</sup> Bonita BPMS: <http://www.bonitasoft.com/>

software livre (sob licença GPL<sup>12</sup>), com uma máquina de execução de processos que oferece mecanismos para extração de informações dos processos instanciados pela organização, além da localidade geográfica em que cada ator da organização atua. A informação da localidade dos atores é importante pelos efeitos que a literatura acadêmica atribui à distância geográfica entre pessoas que precisam colaborar para realizar suas atividades, conforme discutido no Capítulo 2, possibilitando que essa informação seja levada em conta na análise de colaboração.

## 4.5 Medição da Colaboração

A etapa seguinte é a de análise da colaboração em um processo, ocorrendo posteriormente à extração dos dados das instâncias de processo. Após a busca na literatura por métricas de colaboração em processos e das informações disponíveis nos *logs*, a análise de redes sociais foi escolhida para medir a colaboração, utilizando para isso as propriedades descritas no Capítulo 2: densidade, intermediação e centralização da rede; e grau de centralidade e intermediação dos nós.

Para conseguir extrair as medidas de redes sociais dos processos, os dados coletados pelo EvolTrack-Process são agrupados para reconstrução de modelos dos processos, juntamente com a reconstituição de como ocorreu a execução da instância. Essa reconstrução é feita sob a forma de um grafo sobre o qual são calculadas as métricas de redes sociais, onde o conjunto dos vértices é formado por todos os atores, atividades, artefatos e elementos de fluxo existentes no processo, enquanto as arestas são criadas quando há relacionamento entre esses elementos: quando um ator participa da execução de uma determinada atividade; entre duas atividades quando uma deve ser realizada após a finalização da outra; entre elementos de fluxo e as atividades que iniciam cada fluxo do processo; e entre artefatos e as atividades que os produziu.

Uma vez reconstruído o processo, surge o desafio de como analisar a colaboração. O campo de análise de redes sociais é rico em métricas derivadas das propriedades matemáticas dos grafos, cada uma delas refletindo uma característica diferente da rede observada, e, para o ambiente EvolTrack-Process, cinco métricas foram escolhidas por proporcionarem indícios mais alinhados aos aspectos de colaboração: densidade, intermediação e centralização da rede; e grau de centralidade e de intermediação dos vértices.

---

<sup>12</sup> GPL (GNU General Public License) é uma licença de software livre. Mais detalhes em: <http://www.gnu.org/licenses/licenses.html#GPL>

As métricas de densidade, intermediação e centralização da rede dão um panorama geral dos relacionamentos dos processos, sendo utilizadas em estudos relacionados para aferir as características de coordenação do processo (COSTA, SANTANA, *et al.*, 2014) (SANTOS, ARAUJO e MAGDALENO, 2011). Essas mesmas medidas foram relacionadas com o nível de maturidade de colaboração do processo observado (COSTA, SANTANA, *et al.*, 2014), utilizando como base o modelo de maturidade CollabMM (MAGDALENO, ARAUJO e WERNER, 2011).

As métricas de grau de centralidade e intermediação dos vértices, por sua vez, são aplicadas aos atores do processo, e indicam quais atores concentram grande parte dos relacionamentos da rede. Essas métricas são importantes para avaliar a importância de cada ator para uma determinada instância, dando indícios também dos atores que estiveram mais presentes que os demais durante a realização de suas atividades. A métrica de grau de centralidade, por exemplo, quantifica o número de relacionamentos de cada ator (i.e. quantas atividades em comum ele realizou com atores distintos), e é importante para reconhecer quais atores colaboraram com o maior número de outros, possibilitando a identificação de atores com perfis de coordenadores da rede social. A métrica de grau de intermediação, por sua vez, determina quão próximo um determinado ator está dos demais (i.e. quantos outros elementos da rede ele precisa acessar para alcançar os demais atores), sendo extremamente importante para saber quem são os atores que conseguem propagar informações mais rápido para os demais.

## 4.6 Visualizações de Colaboração

Um dos diferenciais do ambiente é a utilização de técnicas de visualização de informação aplicadas a processos de software. Embora vários trabalhos apliquem técnicas de visualização para o desenvolvimento de software em geral – originando a subárea denominada Visualização de Software, descrita no Capítulo 3 – a utilização de visualizações no contexto de processos não é comum, e as soluções encontradas (KANG e HAN, 2008) (IBM, 2003), em geral, adotam gráficos simples utilizados apenas para compor relatórios, como o *dashboard* oferecido pela CLM (*Collaborative Lifecycle Management*) da iniciativa IBM Jazz (SACHDEV, 2013), que disponibiliza gráficos de linha e de colunas.

O volume pequeno de trabalhos relacionados que apliquem visualizações à processos torna difícil a escolha de quais técnicas empregar, uma vez que não há evidências de quais delas seriam adequadas a esse contexto. Nesse sentido, quanto maior a proximidade das visualizações com os modelos e metáforas visuais já

existentes na realidade de trabalho dos usuários, melhores são as chances de interpretação das visualizações (MASON, 2004).



Figura 4.5. Exemplo de gráficos exibidos em ferramentas de acompanhamento de processos (KANG e HAN, 2008).

Na área de processos, o uso de modelos é antigo e bastante comum para representação das informações e planejamento de atividades, com várias notações tendo sido criadas com essa finalidade (SCACCHI, 2001). Seguindo o raciocínio de que analogias com metáforas do cotidiano dos usuários auxiliam na interpretação dos dados, adotar uma **visualização que utiliza uma notação de processos para representar o processo reconstruído** a partir dos *logs* pode reduzir a dificuldade de interpretação das informações exibidas. Como o foco do ambiente é auxiliar em processos de software, a notação definida pelo metamodelo *Software & System Process Engineering Meta-Model* (SPEM) (OBJECT MANAGEMENT GROUP, 2008) foi escolhida como mais apropriada para a representação, já que se trata de uma notação criada especificamente para o contexto de desenvolvimento de software, definida como padrão pelo *Object Management Group* (OMG)<sup>13</sup> e amplamente utilizada por ferramentas de apoio à processos de software (BENDRAOU, JEZEQUEL, *et al.*, 2010).

Contudo, apenas a reconstrução do processo utilizando uma notação criada especificamente para modelar o desenvolvimento de software não é suficiente para tornar evidente a colaboração que ocorreu durante a execução de uma instância. Para obter uma percepção mais acurada da colaboração, é preciso apresentar informações

<sup>13</sup> <http://www.omg.org/>

mais detalhadas em relação as entidades que efetivamente colaboraram durante a produção do software, ou seja, dos atores que participaram do desenvolvimento.

Nesse sentido, algumas informações relacionadas ao desenvolvimento de software não fazem parte dos elementos básicos das notações de processos existentes e, portanto, não aparecem nos modelos utilizados pela indústria. Especificamente no contexto de colaboração, uma dessas informações que foi incorporada ao ambiente EvoITrack-Process é a que diz respeito à localidade geográfica dos desenvolvedores, que pode auxiliar no entendimento e planejamento do desenvolvimento a partir do conhecimento do pessoal disponível e seus respectivos papéis exercidos nos diversos sítios de desenvolvimento (HERBSLEB e GRINTER, 1999) (HERBSLEB, MOCKUS, *et al.*, 2001). Para visualizar essas informações foi planejado outro requisito do ambiente, determinando a construção de um grafo sobre uma projeção do globo terrestre para exibir simultaneamente a rede de relações entre os atores e a localidade geográfica dos mesmos.

Em relação à variabilidade dos processos, o EvoITrack-Process assume que mesmo depois de seguir toda a metodologia proposta pela COMPOOTIM, que assiste na criação de um processo ajustado para maximizar a colaboração diante das características do projeto, a diversidade em processos de software é inevitável (SIEBEL, COOK, *et al.*, 2003). Portanto, é esperado que existam fluxos opcionais para garantir que o processo possa se adequar aos vários contextos de desenvolvimento que um produto pode passar. Nesse sentido, foi determinado que outro requisito do ambiente seria proporcionar ao usuário a visão geral dos fluxos das instâncias de processo, possibilitando investigar como as instâncias são conduzidas com maior frequência.

Para destacar essa variabilidade, foi adotada a utilização de cores para realçar fluxos mais comuns nas instâncias coletadas utilizando estatística descritiva, isto é, técnicas estatísticas de sumarização de dados. Para tornar a visualização ainda mais significativa, também foram disponibilizados filtros que podem interferir no processamento e visualização das informações, limitando, por exemplo, as instâncias que tiveram a participação de determinados atores, possibilitando investigar como as instâncias de processo costumam se comportar de um modo geral, respondendo perguntas do tipo quais atividades um determinado ator costuma realizar ou se existe um fluxo de execução mais comum quando uma determinada equipe está presente.

Todas as técnicas de visualizações descritas anteriormente e as justificativas que embasaram suas concepções foram determinantes para estabelecer mais um requisito do ambiente: permitir que usuários possam acompanhar visualmente a participação de atores nas instâncias de processo extraídas.

## 4.7 Considerações Finais

Esse capítulo descreveu os requisitos que nortearam a construção do ambiente EvolTrack-Process, visando complementar as abordagens para lidar com processos de software concebidas no Projeto CDSOFT, em particular a sistemática COMPOOTIM. Foram apresentadas as três etapas de funcionamento planejadas para o ambiente, sendo elas: extração de dados, medição da colaboração e visualizações de colaboração.

Considerando as revisões da literatura apresentadas nos Capítulos 2 e Capítulo 3, o ambiente EvolTrack-Process surge para preencher uma lacuna observada nos trabalhos de análise de colaboração no desenvolvimento de software. Por um lado, várias abordagens que trabalham a colaboração foram encontradas, mas elas ignoram dados de processo e, portanto, não oferecem informações importantes relacionadas ao planejamento das atividades, fluxo adotado, entre outras. As soluções para acompanhamento de processos encontradas, por sua vez, se concentram em apresentar relatórios sobre o andamento do processo em termos de desvios de estimativa, número de atividades pendentes, entre outras informações, mas não oferecem facilidades para acompanhar a atuação dos atores ao longo do desenvolvimento, ou seja, com quem trabalharam e onde estiveram, etc. Finalmente, trabalhos de mineração de processos em geral extraem as mesmas informações que o ambiente EvolTrack-Process disponibiliza aos usuários, mas se concentram em encontrar relação entre os dados de forma automática, diferentemente do ambiente descrito nessa dissertação, de apresentar visualizações interativas para que os gestores de processos possam tomar suas próprias conclusões com base nos dados e sua experiência.

Os requisitos apresentados nesse capítulo guiaram o desenvolvimento de um protótipo do ambiente, cuja concretização técnica é descrita no próximo capítulo.



## 5 Protótipo

*Este capítulo detalha o protótipo desenvolvido nessa dissertação, que incorpora as funcionalidades discutidas no capítulo anterior para acompanhamento da colaboração. São descritos os requisitos, detalhes sobre a arquitetura, soluções utilizadas e funcionalidades oferecidas. Ao final, um exemplo de uso criado para ilustrar como o protótipo pode ser utilizado é apresentado.*

### 5.1 Introdução

Seguindo o direcionamento oferecido no capítulo anterior, foi desenvolvido um protótipo que incorporou as funcionalidades propostas para o ambiente EvoTrack-Process, de modo que os conceitos e mecanismos propostos para lidar com a colaboração fossem concretizados e a hipótese desse trabalho pudesse ser avaliada.

O ambiente EvoTrack-Process está inserido dentro do contexto do ambiente EvoTrack, que tem por objetivo o rastreamento e visualização da evolução de projetos de software sob diferentes perspectivas (WERNER, MURTA, *et al.*, 2011). O ambiente EvoTrack foi desenvolvido e é mantido pelo grupo de reutilização da COPPE/UFRJ<sup>14</sup>, consistindo em um *plug-in* para a IDE Eclipse capaz de coletar e exibir dados de projetos de software a partir de diferentes fontes de dados (e.g., sistemas de controle de versão), com o propósito de permitir que usuários compreendam como o software foi se modificando e evoluindo com o passar do tempo. O ambiente EvoTrack também conta com uma arquitetura extensível, facilitando a criação de outros *plug-ins* que expandem as funcionalidades do ambiente original, como o EvoTrack-SocialNetwork, *plug-in* voltado para a visualização de redes sociais a partir de artefatos de código-fonte (VAHIA, MAGDALENO e WERNER, 2011; MAGDALENO, ARAUJO e WERNER, 2012).

Esse capítulo apresenta detalhes técnicos sobre a construção e implementação do ambiente EvoTrack-Process, e está organizado da seguinte forma: a Seção 5.2 apresenta o projeto técnico do ambiente, descrevendo requisitos e a arquitetura proposta. A Seção 5.3.1. descreve o próprio protótipo desenvolvido, incluindo as visualizações e mecanismos implementados para lidar com os dados exibidos. A Seção 5.4 apresenta demonstrações das funcionalidades do ambiente utilizando um

---

<sup>14</sup> <http://reuse.cos.ufrj.br>

cenário fictício. Finalmente, a Seção 5.5 encerra o capítulo, apresentando as considerações finais.

## 5.2 Projeto Técnico

O projeto técnico do ambiente EvolTrack-Process teve como objetivo complementar a infraestrutura já oferecida pelo ambiente EvolTrack, oferecendo aos usuários a coleta de dados automatizada e visualizações interativas acerca de processos de software com foco em colaboração.

As próximas subseções apresentam os requisitos e a arquitetura elaborada para atender esses requisitos.

### 5.2.1 Requisitos

Com base nas revisões de literatura, juntamente com as características levantadas no Capítulo 4, foi observado que um ambiente voltado para a visualização da colaboração em processos de software deve contemplar os seguintes requisitos:

**Req01: Extração automatizada de dados.** Dada a especificação do usuário, incluindo o tipo e endereço do repositório, credenciais de acesso, e instâncias a serem analisadas, o ambiente deve coletar automaticamente todos os dados que se encaixem no perfil, sem a necessidade de maior interação ou acompanhamento por parte do usuário, reutilizando os mecanismos disponíveis na infraestrutura do ambiente EvolTrack. Os dados específicos dessa dissertação coletados são informações sobre os atores (incluindo sua localidade), artefatos e atividades de processos incluindo seus relacionamentos, ou seja, em que atividade o artefato foi produzido e em quais atividades os atores participaram. Além dessas informações, o ambiente também deve extrair identificadores do processo e das instâncias executadas.

**Req02: Processamento incremental.** O ambiente deve ser capaz de oferecer aos usuários a opção de "alcançar" as instâncias de processo mais recentes, coletando dados de processos não coletadas ou ainda em andamento. Dessa forma, o ambiente deve ser capaz de integrar dados processados previamente com dados de novas requisições solicitadas pelo usuário.

**Req03: Suporte para extração de dados em diferentes repositórios.** Existem diversas fontes de dados de processos disponíveis, com seus mecanismos de acesso e estruturas de dados particulares. O ambiente deve oferecer facilidades para a inclusão de novos extratores de dados para que possa evoluir e se manter útil, como por exemplo em *logs* de processos disponibilizados em ferramentas de execução de processos.

**Req04: Permitir que o usuário acompanhe a participação de atores em instâncias de processo.** O ambiente deve ser capaz de exibir graficamente os dados coletados e processados através de técnicas de visualização aplicadas à processos, mostrando todos os artefatos produzidos, atores envolvidos e em quais atividades eles atuaram ou estão atuando nas instâncias de processo observadas. As visualizações devem apresentar: um modelo de processo, linguagem comumente utilizada pelo público da ferramenta; uma rede social, construída a partir da colaboração entre os atores que participaram das mesmas atividades; e a apresentação de propriedades de redes sociais sob a forma de uma tabela. Todas as visualizações devem ser interativas, possibilitando ao usuário ocultar/exibir informações que não são relevantes para a análise em andamento, como artefatos ou um conjunto particular de atores.

**Req05: Permitir que o usuário tenha uma visão geral dos fluxos que as instâncias de um processo seguiram ao longo do tempo.** Para auxiliar o usuário no entendimento de como as instâncias se comportam de modo geral ou vem evoluindo ao longo do tempo, o ambiente deve ser capaz de exibir os fluxos mais comuns que as instâncias tem seguido de forma gráfica, usando uma escala de cores para tornar a interpretação intuitiva. Essa informação tem como objetivo auxiliar a identificação de padrões de colaboração entre as instâncias de processo já executadas, possibilitando ao usuário ver, por exemplo, como as instâncias foram conduzidas com mais frequência quando um determinado ator ou equipe participou do desenvolvimento.

**Req06: Oferecer visão de localidades.** Um desafio para a colaboração é a dispersão geográfica dos atores. O ambiente deve ser capaz de exibir informações georeferenciadas com base na localização disponível na base de processos, possibilitando ao usuário identificar onde estão cada um dos atores que participam ou participaram da instância de processo visualizada. Essa informação deve ser exibida sob a forma de um grafo sobre um mapa do globo terrestre para referência espacial.

## 5.2.2 Arquitetura

O projeto de arquitetura do EvolTrack-Process segue a estrutura definida pelo projeto EvolTrack, dividido em quatro módulos principais: *kernel* (núcleo); coleta de dados; transformador de modelos; visualização de processos. Além desses, foi projetado um módulo novo específico para o ambiente dessa dissertação, sendo este o módulo de análise da colaboração (Figura 5.1).

O *kernel* é um módulo provido inteiramente pelo projeto EvolTrack, responsável pela orquestração das operações do sistema e todos os seus *plug-ins*, incluindo a EvolTrack-Process, e atua coordenando os demais módulos e executando

chamadas na seqüência correta. Como um plug-in para a IDE Eclipse, o *kernel* especifica pontos de extensão e classes abstratas relacionadas a implementação de cada um dos outros módulos, conferindo extensibilidade ao projeto.

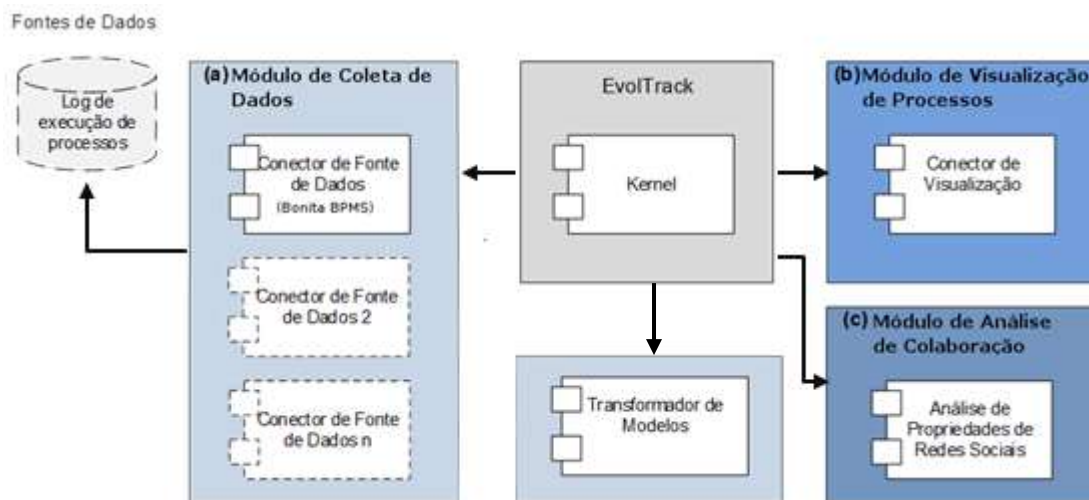


Figura 5.1. Visão geral da arquitetura do ambiente EvoTrack-Process.

O **módulo de coleta de dados** (Figura 5.2) é responsável pela conexão, autenticação, extração e persistência de informações de repositórios de software em um sistema de gerenciamento de banco de dados relacional, ampliando as estruturas presentes no ambiente EvoTrack. A opção de adotar um banco de dados relacional diante de outros não-relacionais (e.g., NoSQL) se justifica dada a maturidade e robustez da tecnologia, que garante uma única linguagem de pesquisa estruturada por todos os sistemas de gerenciamento de banco de dados (SGBD) relacionais através do padrão SQL, o que facilita a migração para outros SGBDs mais modernos no futuro e simplifica o desenvolvimento de novos módulos por outros desenvolvedores, uma vez que empregam uma linguagem bem conhecida para a realização de consultas e operações (VICKNAIR, MACIAS, *et al.*, 2010).

O módulo foi projetado para facilitar a inclusão de novos conectores conforme estabelecido pelo requisito Req03, adotando padrões de projeto e classes abstratas para conferir extensibilidade ao módulo (GAMMA, HELM, *et al.*, 1995), tornando mais fácil a adição de novas fontes de dados. Dessa forma, diante da necessidade de implementação de outros conectores no futuro, se faz necessário apenas que o desenvolvedor crie uma nova especialização da classe abstrata *ProcessDataCollector*, implemente os métodos abstratos adequadamente, e crie novos valores de constantes associados ao tipo de conector criado.

Para garantir uniformidade entre os diferentes repositórios, foi utilizado o modelo de dados descrito no Capítulo 4, representado no diagrama pelo pacote

entities. O modelo exige que o desenvolvedor converta os dados de processos coletados em uma estrutura comum composta por quatro elementos: atividades, artefatos, atores e elementos de fluxo (estados iniciais/finais, pontos de paralelismo ou de decisão).

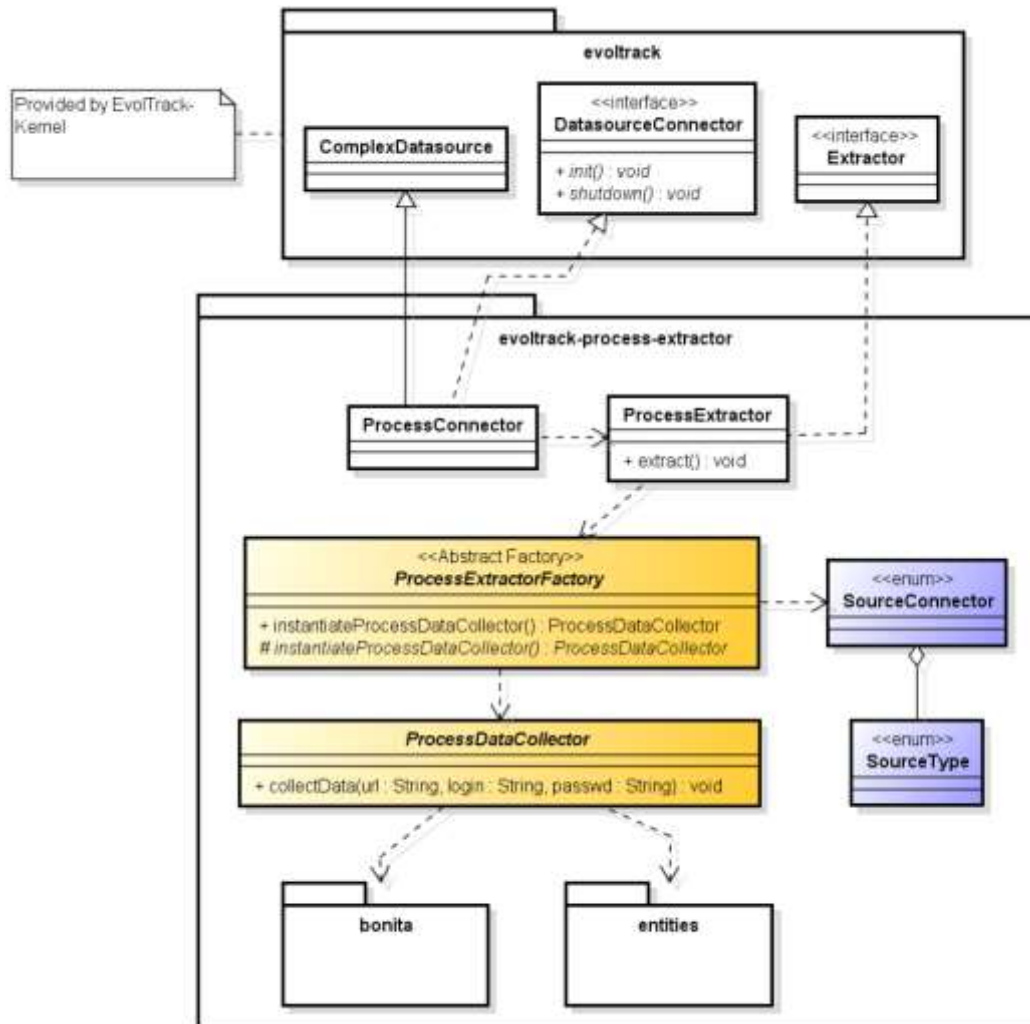


Figura 5.2. Diagrama de classes do módulo de coleta de dados.

O **módulo de transformação** de dados converte toda a informação coletada e persistida no sistema de banco de dados em um modelo UML, a estrutura básica que o *kernel* utiliza para trabalhar com as estruturas diferentes que cada *plug-ins* utiliza. O *EvoTrack-Process* converte as informações em um diagrama de classes, e, portanto, devem ser convertidas em elementos desse diagrama: classificadores (classes, interfaces, *enumerations*), atributos, operações e associações. Diante dessa limitação, o módulo converte as informações de processo nas seguintes estruturas: as atividades do processo são convertidas em interfaces; atores em classes; e estados iniciais, finais, pontos de paralelismo, decisão e artefatos em enumerações (*enumeration*) com um atributo constante que os identifiquem.

Do ponto de vista técnico, o módulo de transformação é o mais simples dos módulos, formado apenas por dois artefatos. Para apoiar essas tarefas, foram reutilizados estruturas de duas extensões do Eclipse, Papyrus<sup>15</sup> e Model Development Tools<sup>16</sup> (MDT). Ambas as extensões tem como objetivo prover ao usuário um ambiente integrado para a construção de modelos – incluindo modelos UML – e foram reutilizados para que não fosse necessário desenvolver essas estruturas e regras de validação de modelos.

O **módulo de visualização de processos** converte as informações de diagramas de classe e gera redes sociais representando os relacionamentos entre as entidades de processo. As visualizações foram construídas com o apoio de uma série de *frameworks* e bibliotecas, a saber: Prefuse, GoogleMaps, D3.js. Essas bibliotecas e *frameworks* foram escolhidos pela facilidade de uso percebida durante a investigação de soluções reutilizáveis, além de terem sido encontrados com frequência em trabalhos similares.

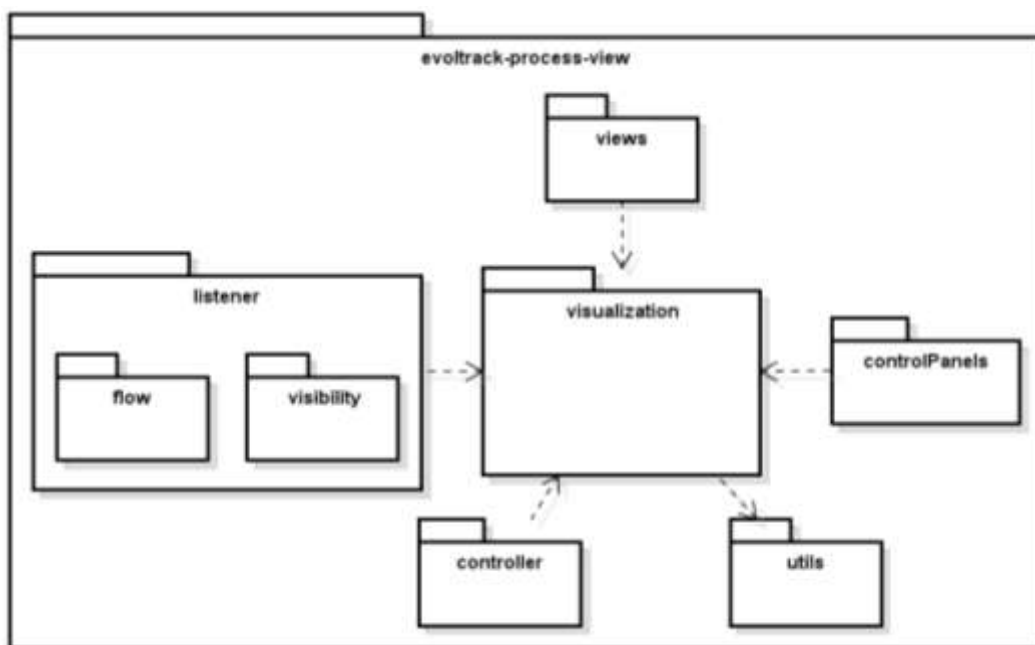


Figura 5.3. Diagrama de pacotes do módulo de visualização.

Além de construir as visualizações, o módulo de visualização de processos também é responsável por: construir a perspectiva e visões do ambiente no Eclipse; apresentar o painel de controle, que permite que o usuário navegue entre diferentes

<sup>15</sup> Disponível em: <http://eclipse.org/papyrus>.

<sup>16</sup> Disponível em: <http://eclipse.org/modeling/mdt/?project=uml2>

instâncias de processos coletadas; e controles e filtros, que permitem que o usuário interfira nas visualizações para se concentrar nos elementos que realmente importam para a análise que estão conduzindo.

Sob o ponto de vista de projeto de solução, cada uma dessas responsabilidades do módulo foi isolada em um pacote (Figura 5.3). O pacote *visualization* concentra as interfaces e classes que definem o funcionamento de uma visualização do ambiente. Filtros e mecanismos de interação são encapsulados pelo pacote *listener*, que captura a interação do usuário e modifica as visualizações, enquanto o pacote *controlPanels* encapsula o controle utilizado pelos usuários para navegar pelas diferentes instâncias do processo. O pacote *controller* implementa rotinas que lidam com a adequação das visualizações quando novos dados são coletados, respondendo ao requisito Req02. Por fim, o pacote *views* é o responsável por montar as visualizações, filtros, mecanismos de interação e painéis de controle na interface da IDE Eclipse, sendo responsável pela interface gráfica de apresentação do ambiente.

Finalmente, o **módulo de análise de colaboração** calcula métricas de análise de redes sociais sobre os grafos. O módulo de análise de colaborações também é simples do ponto de vista técnico, e converte o modelo de processos em um outro grafo, dessa vez utilizando o framework JUNG<sup>17</sup>, para análise de métricas. Embora o framework Prefuse seja mais rico sob o ponto de vista de mecanismos voltados para visualizações, o JUNG disponibiliza a avaliação das propriedades de redes sociais exigidas por esse trabalho – centralidades de grau, proximidade e intermediação, centralização e densidade – sendo mais completo em relação ao tratamento de medidas de grafos.

### 5.3 Protótipo EvoTrack-Process

O protótipo EvoTrack-Process exibe todos os dados coletados em uma mesma tela, oferecendo múltiplas visões para que o usuário possa combinar as diferentes visualizações em suas análises (Figura 5.4. Tela do protótipo EvoTrack-Process exibindo dados de uma instância de processo.).

As visualizações se tornam disponíveis uma vez que dados de processos são coletados. A Figura 5.5 exibe o módulo de coletas de dados, acessado através das preferências da IDE Eclipse, e requer que usuários escolham a fonte de dados de processos, credenciais de acesso, nome de projeto para identificação e instâncias a

---

<sup>17</sup> Disponível em: <http://jung.sourceforge.net/>

serem coletadas, com a opção de buscar todas as instâncias disponíveis no endereço informado. O módulo de extração de dados implementado atende aos três requisitos especificados para o ambiente associados à coleta especificados para o ambiente: Req01, Req02 e Req03.

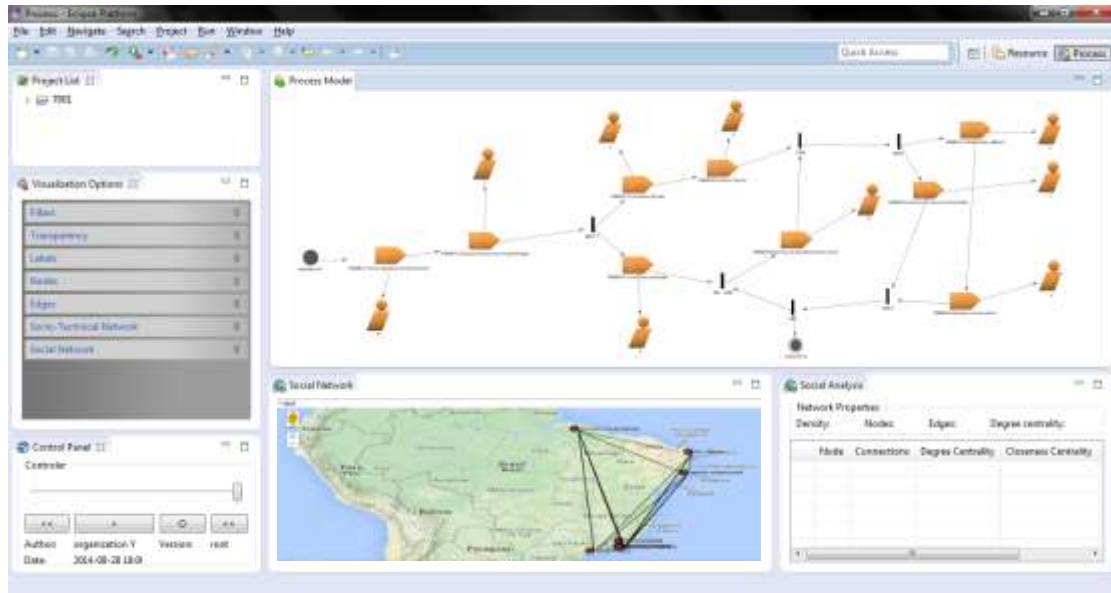


Figura 5.4. Tela do protótipo EvoTrack-Process exibindo dados de uma instância de processo.



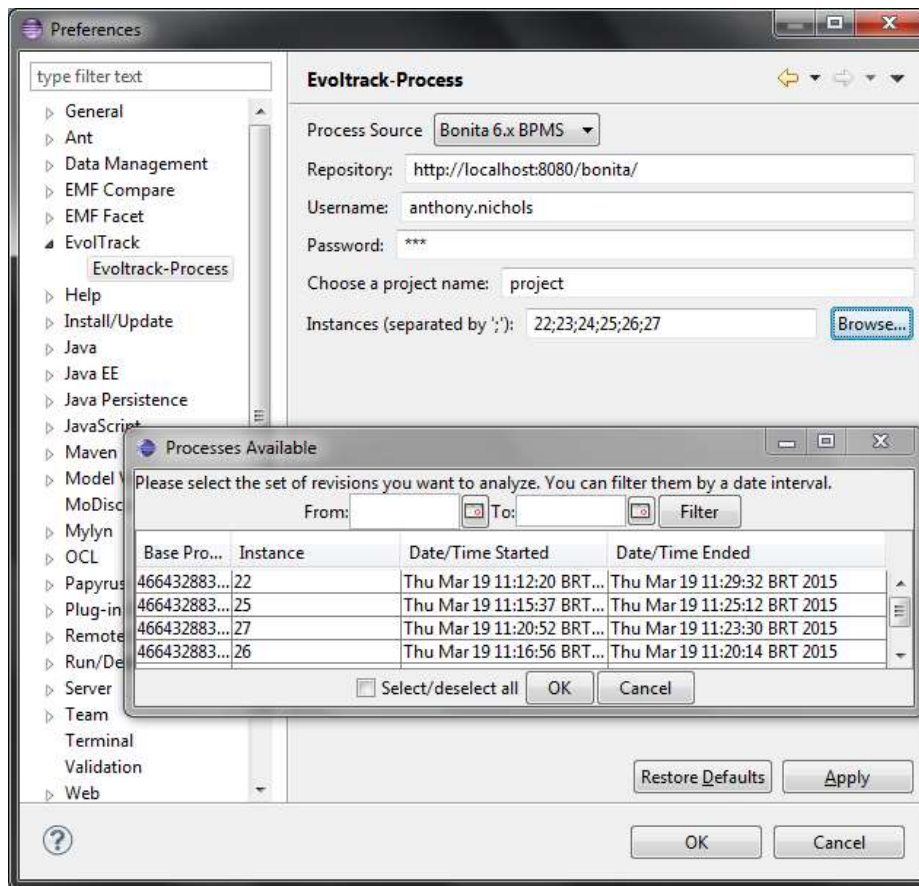


Figura 5.5. Tela de coleta de dados do ambiente EvolTrack, exibindo o conector desenvolvido para o EvolTrack-Process.

As próximas subseções detalham as visualizações e os mecanismos de interação disponíveis no ambiente.

### 5.3.1 Visualizações e Mecanismos de Interação

De acordo com os requisitos especificados para o ambiente, foram desenvolvidas três visualizações para auxiliar nas tarefas de acompanhar e monitorar a colaboração durante o desenvolvimento de software, usando como base dados de processos. As visualizações são descritas a seguir, e foram projetadas para serem visualizadas em monitores e telas convencionais, atendendo ao requisito Req04 do ambiente.

**Modelo do processo.** Como mencionado no Capítulo 4, o uso de modelos é bastante comum para a representação das informações e o planejamento de atividades (SCACCHI, 2001), e, portanto, foi adotado como base para a visualização de processos, utilizando uma metáfora visual presente no cotidiano dos usuários.

O modelo de processo exibido pelo ambiente EvolTrack-Process mostra todas as atividades presentes no modelo do processo, inclusive aquelas que não foram

executadas (representadas pela cor cinza), além dos atores que participaram da execução e dos elementos de fluxo presentes. A visualização foi construída utilizando o framework Prefuse<sup>18</sup>, com ícones para os elementos de processo refletindo os propostos pela notação SPEM (Figura 5.6).

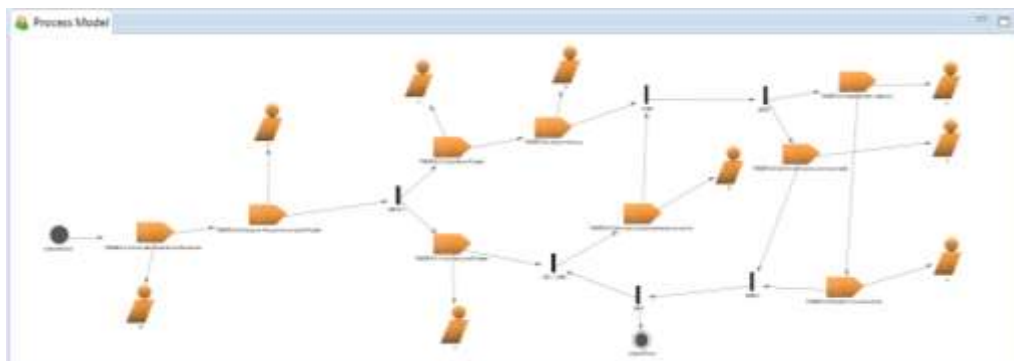


Figura 5.6. Visualização de um processo no ambiente EvolTrack-Process.

**Grafo georreferenciado.** Sistemas que disponibilizam informações geográficas se tornaram muito populares na década de 2000, sobretudo com o advento de *smartphones*, oferecendo receptores GPS (*Global Positioning System*) para bilhões de pessoas ao redor do mundo (VON WATZDORF e MICHAHELLES, 2010). Exemplo desses sistemas são o GoogleMaps<sup>19</sup> oferecido pela Google, que apenas dois anos depois de seu lançamento já havia oferecido seus serviços à uma comunidade de aproximadamente 71,5 milhões de usuários (HAKLAY, SINGLETON e PARKER, 2008); e o OpenStreetMap<sup>20</sup>, uma base de dados geográficos aberta onde pessoas podem contribuir e mapear lugares em qualquer ponto do globo terrestre.

A construção do grafo georreferenciado disponível nessa visualização foi construída com apoio dos serviços gratuitos GoogleMaps e OpenStreetMap em etapas distintas.

O primeiro passo para representação geográfica de informações consiste na captura da localização das informações através de um processo denominado de geocodificação, no qual endereços são convertidos em coordenadas geográficas (latitude e longitude). Esse passo foi realizado utilizando o OpenStreetMap através de consultas web dos endereços dos membros da organização disponíveis na fonte de dados.

---

<sup>18</sup> Disponível em: <http://prefuse.org/>

<sup>19</sup> Disponível em: <http://maps.google.com>.

<sup>20</sup> Disponível em: [www.openstreetmap.org](http://www.openstreetmap.org).

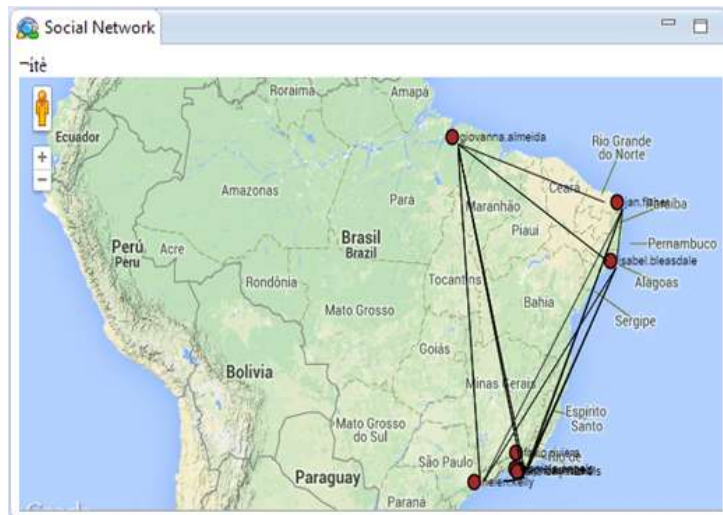


Figura 5.7. Visualização da rede sociotécnica.

Uma vez obtidas as coordenadas dos atores, utilizou-se a API JavaScript GoogleMaps para desenho do mapa interativo. O GoogleMaps foi escolhido por ter exibido um comportamento mais estável e documentação mais detalhada que a fornecida pelo OpenStreetMap. Para desenhar os vértices e arestas do grafo sobre a página web gerada pelo GoogleMaps, foi utilizada a biblioteca JavaScript D3.js<sup>21</sup>, que oferece uma série de visualizações voltadas para serem executadas em navegadores web.

O resultado final da visualização está exemplificado na Figura 5.7. Visualização da rede sociotécnica., onde cada vértice representa um membro da organização, e as arestas interligam aqueles que atuaram juntos durante a execução de uma determinada atividade, possibilitando compreender os relacionamentos entre diferentes sítios de desenvolvimento em uma única imagem, atendendo ao requisito Req06 do ambiente.

**Tabela de métricas.** Embora várias propriedades de redes sociais sejam intuitivas através da simples observação de uma rede, as medidas obtidas pelo módulo de avaliação da colaboração (descritas no Capítulo 4) são disponibilizadas sob a forma de uma tabela, possibilitando a comparação entre nós em casos onde a visualização não é clara e pode induzir interpretações erradas (Figura 5.8).

<sup>21</sup> Disponível em: <http://d3js.org>

The screenshot shows a window titled 'Social Analysis' with a 'Network Properties' section. Below this, a table lists metrics for 8 nodes. The 'Density' is 0.786, 'Nodes' are 8, and 'Edges' are 44. The table columns are Node, Connections, Degree Centrality, Closeness Centrality, and Betweenness.

Node	Connections	Degree Centrality	Closeness Centrality	Betweenness
anthony.nichols	0	0.0	0.0	0.0
giovanna.alm...	7	1.0	1.0	56.0
jan.fisher	7	1.0	1.0	56.0
favio.riviera	7	1.0	1.0	56.0
april.sanchez	7	1.0	1.0	56.0
helen.kelly	7	1.0	1.0	56.0
isabel.bleasdale	7	1.0	1.0	56.0
daniela.angelo	7	1.0	1.0	56.0

Figura 5.8. Visualização de métricas.

### 5.3.2 Filtros e Mecanismos de Interação

Para contornar o problema da sobrecarga de informações discutido no Capítulo 3, é necessário a utilização de mecanismos eficientes de tratamento de dados e informações (e.g., classificação e filtragem) (CHEN, 2006). Esses mecanismos são aplicados às abstrações visuais, possibilitando a interação e a manipulação do que está sendo observado por parte do usuário.

O ambiente EvolTrack-Process oferece ao usuário três tipos de mecanismos de interação: *zoom*, filtros de visibilidade e a visualização de processos mais comuns descritos a seguir.

**Zoom.** O *zoom* é um tipo de filtro de informações que possibilita a ampliação ou redução do tamanho de algum elemento representado graficamente conforme a interação do usuário (*zoom-in* e *zoom-out*) (DIEHL, 2007).

No ambiente EvolTrack-Process o *zoom* é particularmente útil na visualização do modelo de processos, dado que o número de elementos em um determinado processo pode ser grande e impossibilitar a compreensão geral com os elementos expostos em uma escala muito pequena. O *zoom* também é disponibilizado no grafo georreferenciado, oferecendo mais detalhes sobre a localidade ou uma visão mais geral da dispersão geográfica dos indivíduos.

**Filtros de visibilidade.** Os filtros de visibilidade servem para que o usuário não esteja limitado apenas a um conjunto de visualizações estáticas, oferecendo liberdade para ocultar informações que sejam irrelevantes para o contexto de uma determinada observação.

Os filtros de visibilidade afetam apenas a visualização de modelo de processo, possibilitando que o usuário torne invisível elementos de processo, como por exemplo um conjunto de atividades.

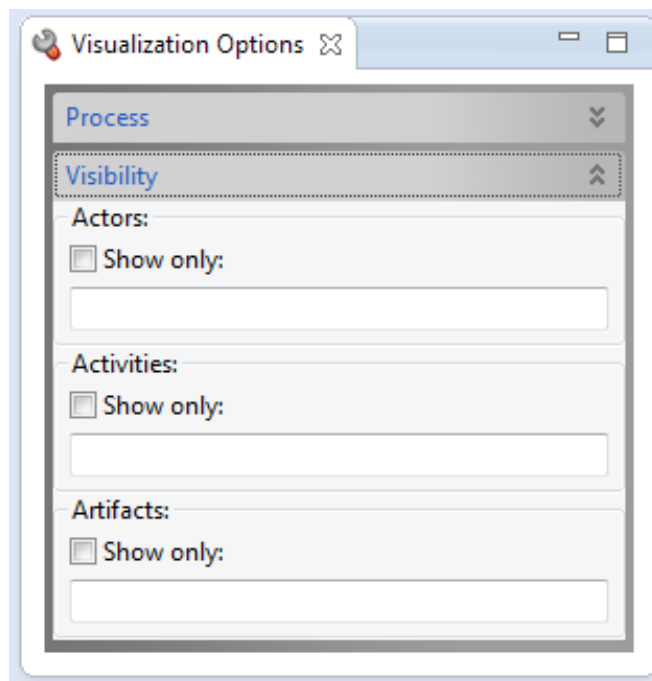


Figura 5.9. Filtros de visibilidade disponibilizados no ambiente EvoTrack-Process.

**Visualização de fluxos de processos mais comuns.** A diversidade em processos de software é inevitável (SIEBEL, COOK, *et al.*, 2003), propiciando casos em que instâncias de um mesmo modelo de processo sigam caminhos de execução diferentes. O ambiente EvoTrack-Process, conforme especificado pelo requisito Req 05, disponibiliza um mecanismo para visualizar quais instâncias de processo ocorrem com mais frequência, oferecendo ao usuário facilidades para entender como a colaboração ocorre nesses casos.

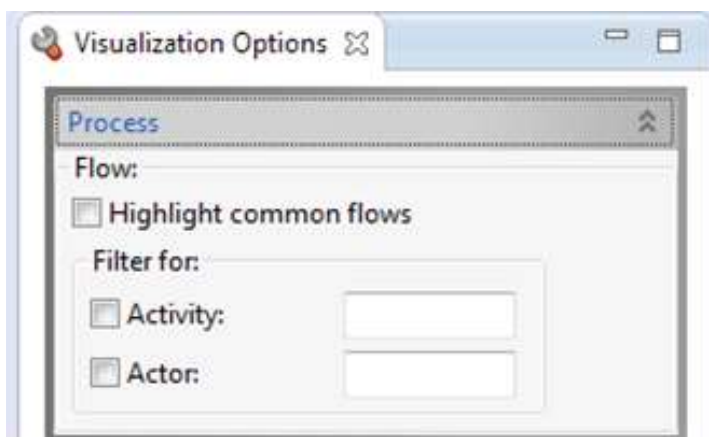


Figura 5.10. Filtro de visualização de fluxos mais comuns de processos.

Os fluxos mais comuns entre as instâncias são definidos a partir do cálculo de quartis, valores de estatística descritiva que dividem um conjunto de dados. O mecanismo divide a ocorrência dos fluxos em quatro grupos: aqueles que quase nunca ocorrem (até 25% dos casos); os que ocorrem em menos da metade (até 50%); os que ocorrem na maioria dos casos (até 75%); e, finalmente, os que ocorrem em todos os fluxos observados (até 100%) (Figura 5.11).



Figura 5.11. Escala de cor utilizada para identificar atividades mais comuns.

O mecanismo de visualização de fluxos de processos mais comuns também afeta apenas a visualização de modelo do processo e sofre interferência dos filtros de visibilidade, possibilitando, por exemplo, visualizar os filtros mais comuns quando um determinado ator participa da execução ou quando uma determinada atividade foi concluída.

## 5.4 Exemplo de Uso

O processo observado neste exemplo é uma versão adaptada de um processo real de desenvolvimento de uma empresa brasileira (Figura 5.12). O processo descreve o tratamento de demandas relacionadas a desenvolvimentos de software, que podem ser aceitas ou rejeitadas segundo critérios utilizados pelo gestor. Uma vez aceita, vários papéis atuam no tratamento da demanda, que culmina no desenvolvimento de um projeto de software.

Foram emuladas oito instâncias executadas por um total de quinze atores separados em duas equipes geograficamente dispersas.

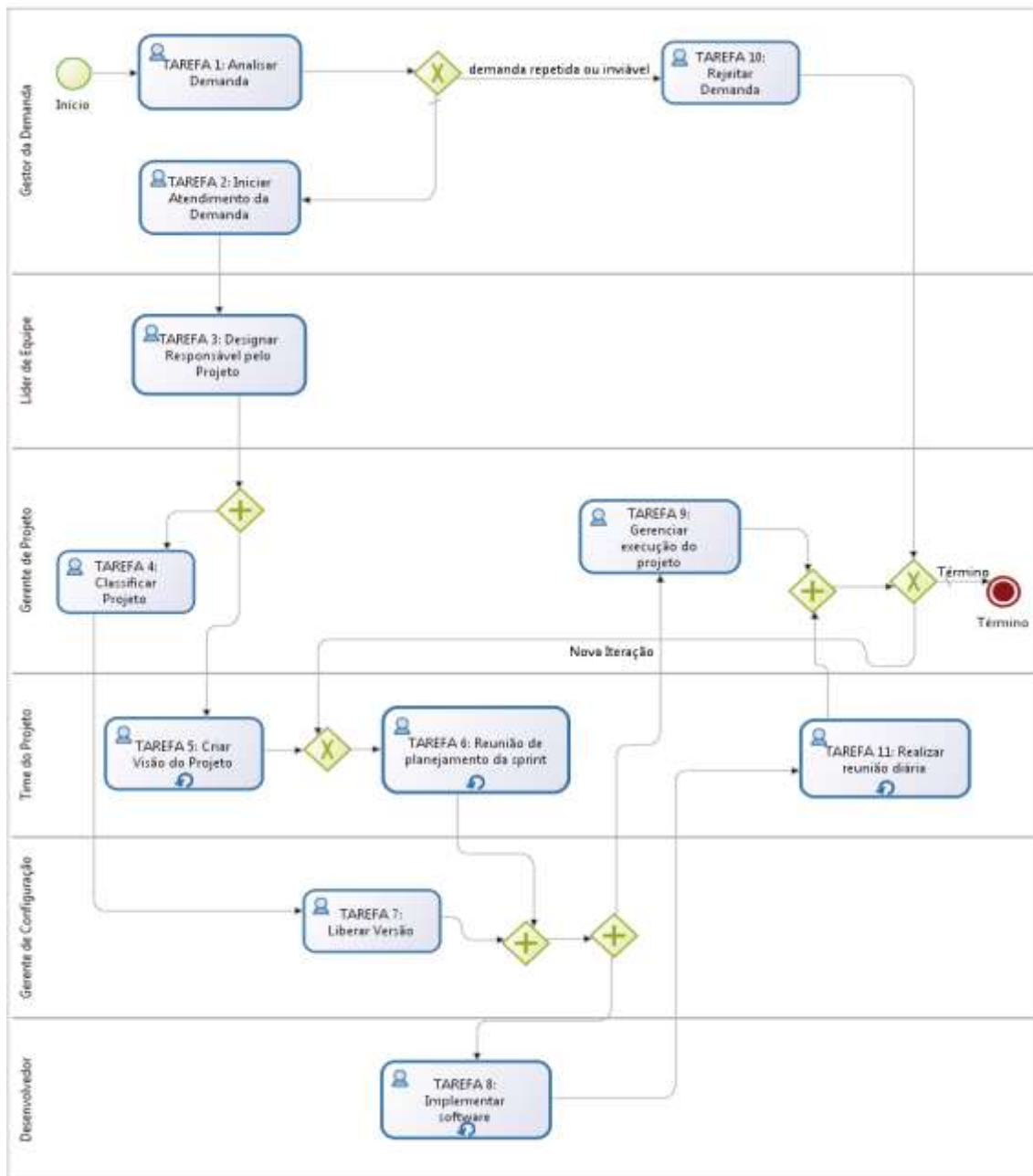


Figura 5.12. Processo de exemplo modelado no Bonita BPMS Studio.

Uma vez coletadas as informações das instâncias, é possível observar várias características das execuções do processo ao coletar os dados utilizando as visualizações disponibilizadas pela EvoTrack-Process. A Figura 5.13 mostra a visualização de uma das instâncias executadas, podendo ser observado que o processo seguiu um determinado fluxo. Observando melhor as atividades através do zoom, constata-se que o fluxo adotado acabou por rejeitar a demanda, o que justifica o grande número de atividades não executadas (cinzas) exibidas.

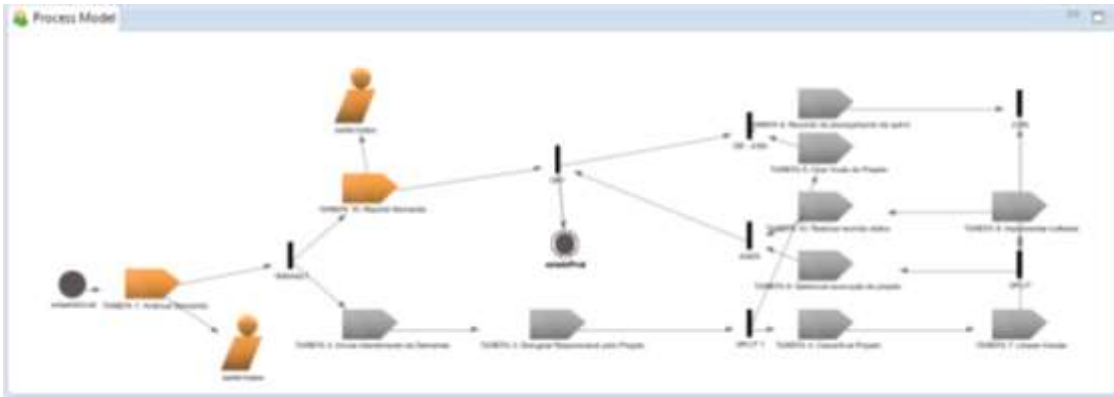


Figura 5.13. Visualização de uma instância executada do processo.

Em outras instâncias onde houve a participação de mais atores, a visualização auxilia na identificação de quem participou da execução de cada uma das tarefas em cada uma das instâncias (Figura 5.14). Com base nessa informação, é possível conhecer atores que costumam trabalhar juntos, o que permite a alocação de um time com um perfil onde os membros já conhecem a forma de trabalho de cada um; identificar potenciais substitutos para a realização de uma atividade diante da indisponibilidade de um determinado ator; entre outras possibilidades. É importante ressaltar que essas informações não necessariamente garantem que haverá melhorias na qualidade das atividades desenvolvidas ou na produtividade dos atores, servindo apenas como apoio adicional para a tomada de decisão por gerentes de processo de software.

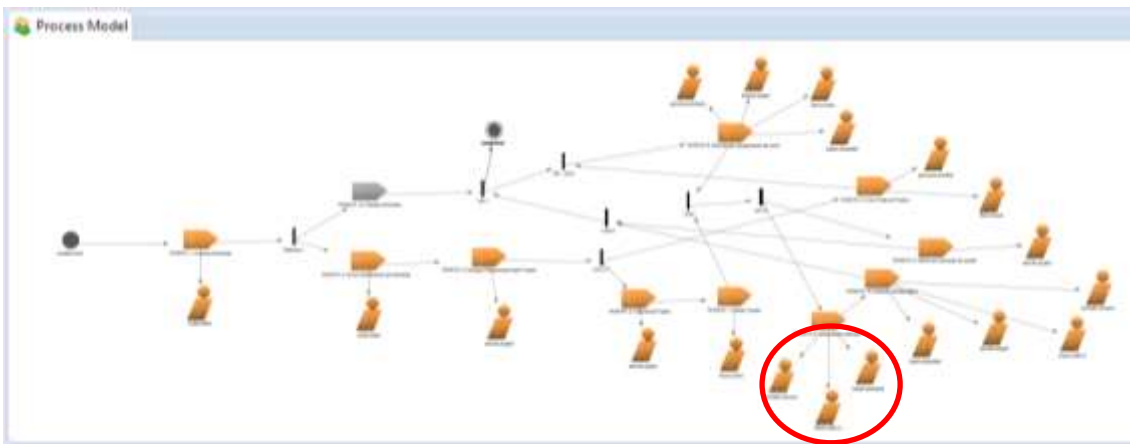


Figura 5.14. Visualização de uma instância do processo, com destaque para vários atores envolvidos na execução de uma mesma atividade.

Os filtros e mecanismos de interação também auxiliam o entendimento de algumas características das instâncias. Por exemplo, em relação aos fluxos mais comuns, é possível observar que a atividade inicial do processo é executada na maioria dos casos, o que já era esperado já que o processo conta com um único



caminho inicial. No entanto, em relação as atividades seguintes, é possível perceber que a maior parte das execuções caminham para um determinado fluxo, que consiste no caminho onde a demanda analisada pelo processo não é rejeitada e ocorre o desenvolvimento do produto (Figura 5.15).

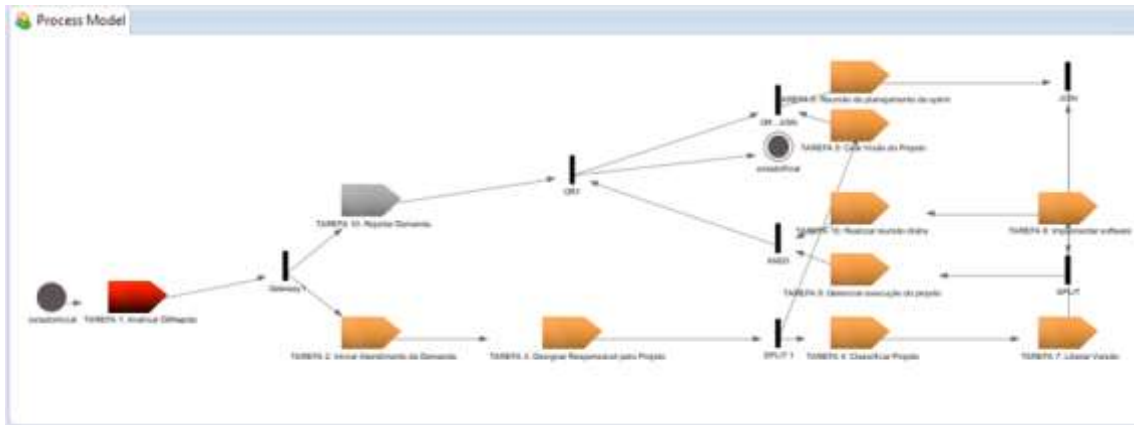


Figura 5.15. Fluxo mais comum de execução do processo.

O quadro muda, porém, quando se considera apenas um dos atores responsáveis pela análise das demandas para a visualização dos fluxos mais comuns. Este ator inicia poucos processos (observado pela cor cinza na primeira atividade), mas o número de demandas rejeitadas nesses casos ocorre tão frequentemente quanto os de demandas aceitas, diferentemente do comportamento geral das instâncias.

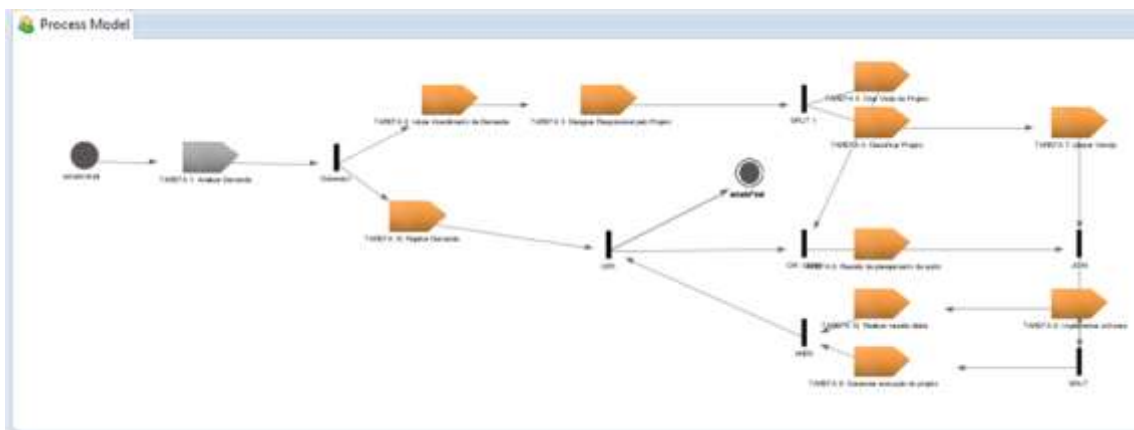


Figura 5.16. Fluxo mais comum de instâncias onde um ator específico participou.

A visualização de grafo georreferenciado, por sua vez, amplia as redes sociais em relação ao que é exibido por várias outras ferramentas, adicionando os dados geográficos dos atores. Essa informação permite identificar quais sítios se comunicaram durante uma determinada instância (Figura 5.17), além de quais atores estiveram envolvidos.

As visualizações também podem ser combinadas para análises mais complexas, possibilitando a identificação de que tipo de atividade cada sítio ficou responsável por realizar, no caso do processo de exemplo utilizado, quais sítios estão rejeitando mais demandas.



Figura 5.17. Visualização de grafo georreferenciado exibindo colaboração entre atores de dois sítios.

Diferentemente das duas visualizações descritas anteriormente, a tabela de métricas é a visualização mais direta oferecida pelo ambiente EvoTrack-Process, indicando os valores das medidas calculadas para uma determinada instância. A interpretação desses valores já foi discutida no Capítulo 2, e refere-se aos vários graus de prestígio que um ator do processo pode ter de acordo com a medida observada.

Nas instâncias emuladas para o exemplo, as medidas da tabela ratificam algumas das características observadas, apontando, por exemplo, valores altos para o grau de intermediação para atores que interligam dois sítios de desenvolvimento.

## 5.5 Considerações Finais

Esse capítulo detalhou o projeto e a implementação do protótipo do ambiente EvoTrack-Process com base nos detalhes descritos no Capítulo 4. O protótipo se beneficiou da infraestrutura oferecida pelo projeto EvoTrack, e foram apresentados os requisitos e as decisões arquiteturais tomadas para satisfazê-los.

Além de detalhes técnicos de projeto, foram descritas as visualizações presentes no ambiente, bem como os filtros e mecanismos de interação disponibilizados nas visualizações, planejados para facilitar a utilização e ampliar o potencial de análise da ferramenta. O capítulo também apresentou alguns exemplos

de uso do protótipo, ressaltando como as visualizações podem ser utilizadas para entender e acompanhar a colaboração em processos de software.

Vale ressaltar que o protótipo implementado também possui uma série de limitações. Além da dependência em relação a IDE Eclipse, até o presente momento a infraestrutura de coleta de dados permite recuperar dados apenas do BPMS Bonita. Em relação as visualizações, embora tenham sido disponibilizados filtros e mecanismos de interação para reduzir o problema da sobrecarga de informação, não fez parte do escopo desse trabalho garantir que essas visualizações e mecanismos desenvolvidos sejam escaláveis, ou seja, capazes de auxiliar no entendimento e acompanhamento da colaboração em situações onde haja um conjunto muito grande de informação para ser observado.

## 6 Avaliação

*Este capítulo apresenta a avaliação do ambiente, realizada através de um estudo sobre o protótipo desenvolvido. Foi realizado um estudo qualitativo, com o propósito de investigar se as técnicas de visualização disponibilizadas no ambiente auxiliavam os participantes do estudo a compreender melhor como a colaboração acontece durante a execução do processo.*

### 6.1 Introdução

A Engenharia de Software Experimental afirma que a validade de qualquer corpo de conhecimento deve ser avaliada para que esse conhecimento possa ser considerado científico (JURISTO e MORENO, 2010). Seguindo estes princípios, o ambiente EvolTrack-Process foi avaliado através de um estudo para verificar se as visualizações oferecidas de fato representavam um diferencial no entendimento e acompanhamento da colaboração em processos para os participantes durante a execução de tarefas predefinidas.

Este capítulo descreve o planejamento, execução e análise dos dados da avaliação do ambiente EvolTrack-Process apresentado no capítulo anterior. O restante do capítulo está organizado como segue: a Seção 6.2 apresenta os detalhes do planejamento do estudo, descrevendo o objetivo e contexto de avaliação; a Seção 6.3 discute as ameaças à validade do estudo e as medidas tomadas para minimizá-las; a Seção 6.4 detalha o procedimento de execução do estudo; e, por fim, a Seção 6.5 apresenta e analisa os resultados que foram alcançados.

### 6.2 Plano de Avaliação

#### 6.2.1 Objetivo

A hipótese geral desse trabalho, conforme definida no Capítulo 1, é de que técnicas de visualização podem auxiliar na compreensão e acompanhamento da colaboração em processos de software. Em particular, nesse trabalho é investigado se as técnicas de visualizações disponibilizadas pelo ambiente EvolTrack-Process (i.e. modelo de processos, redes sociais georeferenciada e a tabela de métricas) auxiliam no entendimento e acompanhamento da colaboração em instâncias de processo de software.

O tipo de estudo adotado para a condução da avaliação é definido na literatura como estudo de observação ou observacional. Estudos observacionais são aqueles onde o participante realiza alguma tarefa enquanto é observado por um experimentador (SHULL, CARVER e TRAVASSOS, 2001). Esse tipo de estudo tem a finalidade de coletar informações sobre como cada uma das tarefas foi realizada e, através dessas observações, obter uma compreensão de como o ambiente é efetivamente utilizado. O estudo também se encaixa sob a definição de estudos *in vitro* (JURISTO e MORENO, 2010), ou seja, ocorreu em um laboratório sob condições controladas utilizando dados fabricados especificamente para o estudo.

O planejamento da avaliação também se baseou em outros estudos com objetivos similares, que investigaram o uso de visualizações de software em outros contextos. Estudos orientados à tarefa, como o descrito nesse planejamento, são muito populares por conseguirem responder se as visualizações conseguem auxiliar na solução dos problemas dos usuários sob o ponto de vista deles (DIEHL, 2007).

Apresentado de acordo com o paradigma GQM (*Goal Question Metric*) (BASILI, CALDIERA e ROMBACH, 1994), o objetivo do estudo é descrito conforme a Tabela 6.1 a seguir.

Tabela 6.1. Definição do objetivo do estudo.

<b>Analisar</b> o uso das visualizações e mecanismos de interação disponibilizados pelo ambiente <i>EvoTrack-Process</i>
<b>Com o propósito de</b> caracterizar
<b>Com respeito ao</b> entendimento e acompanhamento da colaboração em processos
<b>Do ponto de vista de</b> gerentes de processo sem conhecimento prévio do modelo de processos utilizado
<b>No contexto de</b> processos de software

## 6.2.2 Contexto

### a) Cenário

O estudo coloca cada participante no papel de um gerente de processo que acaba de ser contratado para uma empresa de desenvolvimento de software. A empresa conta com vários sítios distribuídos, e o participante precisa nesse primeiro momento entender como os sítios colaboram entre si e como os times se comportam durante a execução das instâncias de processo de software.

A empresa conta com várias instâncias de processos já finalizadas, podendo ser utilizadas pelos pesquisadores como base para: i) adquirir o conhecimento sobre a

colaboração nas instâncias de processo (entender a colaboração); e ii) investigar como essas instâncias foram variando e os times foram colaborando ao longo do tempo (acompanhar a colaboração).

### **b) Participantes**

O perfil dos participantes não foi restrito à membros da indústria ou academia. O estudo requer que todos os participantes tenham alguma experiência com gestão de processos de software, mas não exige algum nível de formação específico ou tempo de atuação mínimo.

Cada participante utilizou o ambiente EvolTrack-Process em sessões individuais. Não houve nenhum tipo de compensação para os participantes.

### **c) Tarefas**

Neste estudo foi definido um conjunto de tarefas a serem realizadas, visando alcançar os objetivos do estudo. As tarefas planejadas para o estudo refletem características associadas à percepção da colaboração entre atores durante o desenvolvimento de software, buscando verificar se o participante consegue identificar, por exemplo, com quais atores um determinado ator colaborou durante a execução do processo ou quais são aqueles atores que mais colaboraram com outros.

As tarefas foram divididas em três grupos, com diferentes níveis de complexidade, baseado no trabalho de Oliveira (2011).

- **Tarefas de Filtragem (TF):** Este grupo consiste de tarefas mais simples, de forma que, se algum participante não estiver apto a realizá-las, o mesmo deve ser desconsiderado na análise dos resultados (por indicar possíveis problemas no entendimento do funcionamento do ambiente ou das tarefas do experimento). Nesse estudo, estas tarefas são:
  - Quais atores participaram da execução das instâncias **3** e **4**? Escreva o nome de cada um deles.
  - Qual ator colaborou com o maior número de outros atores na instância **5**?
- **Tarefas Básicas (TB):** Este grupo consiste de tarefas básicas, que podem ser realizadas por meio de fatos extraídos a partir do ambiente de processos. Nesse estudo, estas tarefas são:
  - Com quais outros atores de sítios diferentes o ator “**Isabel Bleasdale**” colaborou na instância **1** do processo? Escreva o nome de cada um deles.

- Quais atores já participaram da execução da atividade “**Implementar Solução**” em conjunto com o ator “**Jan Fisher**”? Escreva o nome de cada um deles.
- Quais atividades diferentes o ator “**Giovanna Almeida**” desempenhou na instância 1 em relação à instância 4?
- **Tarefas de Assimilação (TA):** Este grupo consiste de tarefas mais difíceis e complexas, exigindo que o participante utilize o conhecimento prévio e o raciocínio de forma a obter o entendimento e interpretar as informações contidas nos instrumentos da avaliação. Nesse estudo, a única tarefa de assimilação é:
  - Considerando a instância 6 do processo, qual ator você indicaria para substituir o ator “**Helen Kelly**” durante a execução da atividade “**Descrever Detalhes Técnicos**”? Por que?

Os participantes não foram comunicados da existência e do objetivo desta divisão de tarefas em grupos, de forma a não influenciar a execução das atividades. Os participantes também foram orientados a descrever o procedimento que utilizaram para responder a cada uma das tarefas, sendo essas informações qualitativas coletadas pelo pesquisador.

#### ***d) Dados Utilizados***

Este estudo emulou a execução de instâncias de um processo real, disponibilizado por uma empresa nacional de desenvolvimento de software. No entanto, apenas o modelo de processo utilizado é real. Todos os dados e atores envolvidos nas instâncias disponibilizadas no estudo são fictícios, garantindo a confidencialidade das informações da empresa.

O processo, cujo modelo pode ser visto no Anexo A, retrata o procedimento para lidar com solicitações de mudanças, desde suas requisições até a disponibilização das novas funcionalidades nos produtos desenvolvidos. Seis papéis diferentes da empresa são utilizados para a execução do processo.

Foram emuladas seis instâncias executadas do processo, com quinze atores fictícios envolvidos na execução das atividades. Os atores foram também divididos em dois times – um com nove e outro com seis atores – espalhados em sítios diferentes, para retratar o contexto de desenvolvimento distribuído de software.

Mais informações sobre as atividades, papéis e execuções do estudo podem ser visualizadas no Apêndice A.

### **e) Instrumentos**

Para este estudo, foram construídos cinco instrumentos que foram entregues impressos aos participantes (ver Apêndice B. 1):

- Termo de consentimento livre e esclarecido;
- Formulário de caracterização do participante;
- Formulário para realização do estudo;
- Questionário para avaliação do estudo;
- Material de treinamento.

O primeiro instrumento é o termo de consentimento livre e esclarecido, que declara ao participante o objetivo do estudo, os limites da sua participação e suas responsabilidades durante a avaliação. O termo tem como objetivo obter a aprovação do participante no seu envolvimento com o estudo, informa que os dados da avaliação não estão sujeitos a serem utilizados para classificar o desempenho dos participantes e garante a confidencialidade dos dados. Este termo foi assinado pelo participante antes do início do estudo.

Em seguida, o participante foi solicitado a preencher um formulário de caracterização 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, também foi feito um breve treinamento com os participantes visando explicar os principais conceitos de redes sociais e a colaboração em processos.

O formulário para a realização do estudo apresenta ao participante esse contexto de trabalho e as instruções para a realização das tarefas. Este formulário foi acompanhado da apresentação dos dados no ambiente. 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, que pretende obter informações qualitativas acerca do estudo do ponto de vista do participante. Este questionário inclui sugestões para o ambiente EvolTrack-Process, indicadores sobre a utilização de visualizações aplicadas à processos, a facilidade de uso do ambiente e considerações sobre o procedimento de avaliação.

## **6.3 Ameaças à Validade**

É comum que haja questões que possam impactar ou limitar a validade dos resultados dos estudos. Estas questões, denominadas ameaças à validade, são



apresentadas a seguir, categorizadas em interna, externa, de conclusão e de construção, conforme proposto por (WOHLIN, RUNESON, *et al.*, 2000).

### **6.3.1 Ameaças à validade interna**

Ameaças à validade interna são influências que não foram observadas ou controladas que podem afetar os resultados do estudo, dificultando o estabelecimento de causalidade entre o tratamento e o efeito medido. Nesse estudo, foram identificadas as seguintes ameaças nessa categoria:

- 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, RUNESON, *et al.*, 2000). A revisão dos instrumentos utilizados pelos outros pesquisadores envolvidos no estudo visa justamente reduzir esta interferência. Outra medida tomada para minimizar essa ameaça foi a condução de um estudo piloto para obter uma impressão inicial das dificuldades e viabilidade do estudo, conduzido com um aluno de pós-graduação. Os dados do piloto não foram considerados durante essa avaliação;
- A comunicação entre os participantes pode conceder alguma vantagem durante a execução das atividades. Para minimizar esse problema, os participantes não foram comunicados de todos os demais que participariam do estudo. Os participantes também foram orientados a não trocar informações a respeito do estudo que participaram;
- O estudo não foi executado no mesmo dia ou em um mesmo ambiente físico por todos os participantes, não sendo possível controlar influências externas (e.g. interrupções) da mesma forma para em todas as sessões do estudo;
- Imprevistos técnicos podem ocorrer durante a execução das atividades pelos participantes (por exemplo, problemas de energia elétrica, acesso à internet e lentidão no ambiente de execução). Nesse sentido, o ambiente foi configurado e testado antes de cada sessão do estudo pelo pesquisador.

### **6.3.2 Ameaças à validade externa**

Ameaças à validade externa são condições que limitam a capacidade de generalização dos resultados para outros contextos fora do ambiente controlado avaliado. Nesse estudo, foram identificadas as seguintes ameaças nessa categoria:

- O estudo utilizou um único processo de desenvolvimento de software adaptado como fonte de dados. Embora tenha sido utilizado um processo real para criar

um ambiente de análise mais próximo de um possível de ser encontrado na indústria de software, é impossível afirmar que o ambiente se comportaria da mesma forma com processos diferentes;

- O estudo também não se propôs a investigar o apoio das visualizações em todas as situações possíveis onde podem ocorrer a colaboração em processos de software. Dessa forma, não é possível afirmar que os resultados encontrados permaneceriam os mesmos em outras tarefas ou cenários.

### **6.3.3 Ameaças à validade de conclusão**

Ameaças à validade de conclusão referem-se a fatos que prejudicam o estabelecimento de relacionamentos estatísticos entre o tratamento e o resultado, afetando a habilidade de chegar a uma conclusão correta dos relacionamentos entre tratamentos e resultados observados. Nesse estudo, apenas uma ameaça de conclusão foi identificada:

- O tamanho da amostra desse estudo é pequeno, impossibilitando a aplicação de análises estatísticas. Por isso, este estudo apresenta uma limitação nos resultados, que serão considerados apenas como indícios.

### **6.3.4 Ameaças à validade de construção**

Ameaças à validade de construção são efeitos que podem impedir que a configuração do estudo reflita bem a construção do relacionamento causa-efeito observado, ou seja, eventos que podem prejudicar a medição correta no estudo.

- Uma possibilidade é que as métricas e perguntas qualitativas selecionadas não sejam bons indicadores para concluir sobre os objetivos do estudo. Para diminuir esse risco, foram escolhidas medidas que visam capturar as informações necessárias para responder às questões do estudo;
- Existe uma proximidade entre alguns dos participantes e o responsável pela execução do estudo. Dessa forma, não é possível afirmar que não houve uma dedicação adicional durante a realização das tarefas ou que as respostas de avaliação do estudo e do ambiente seriam as mesmas em outros cenários de execução;
- Os perfis dos participantes é bastante heterogeneo, com variações tanto no sentido de grau como no tempo de experiência nos temas relacionados a esse estudo, e não foram estabelecidos pesos para valorizar observações de participantes mais experientes diante de outros com menor experiência. Sendo assim, embora as sugestões, críticas e dificuldades dos participantes tenham

sido apresentadas de forma agregada, não é possível afirmar que alguns desses achados possam ter fundamentação maiores que outras.

- 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 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.

## **6.4 Execução do Estudo**

### **6.4.1 Estudo Piloto**

O estudo piloto foi executado com um participante, e teve como objetivo avaliar as tarefas, treinamento e demais instrumentos criados para a realização do estudo. O participante foi selecionado por conveniência, e é aluno de doutorado do curso de Engenharia de Sistemas e Computação da COPPE/UFRJ.

O participante foi informado sobre o estudo através do Formulário de Consentimento, e, ao concordar em participar no estudo, recebeu em seguida o Formulário de Caracterização. Nesse último, que não continha informações referentes ao grau de conhecimento/experiência com cenários de desenvolvimento distribuído de software, o participante alegou o grau mais alto de experiência nas áreas de conhecimento do formulário, com 36 meses de experiência com Processos de Software, 12 meses na área de Colaboração e 24 meses na área de Visualização de Software. O participante também alegou o grau máximo de experiência com ambientes de acompanhamento de processos.

Após responder o Formulário de Caracterização, o participante recebeu uma sessão de treinamento de aproximadamente 10 minutos sobre as funcionalidades do ambiente EvoTrack-Process e o processo observado. Ao final do treinamento, foi entregue o Formulário para Realização do Estudo, descrevendo o que o participante deveria responder. Durante a execução das tarefas, o participante foi incentivado a comunicar como pretendia responder a tarefa antes de realizá-la, além de indicar passo-a-passo as ações que eram tomadas. O pesquisador tomou nota das informações do participante, além de outras notas gerais relacionadas à observação de como o participante utilizou o ambiente EvoTrack-Process.

O participante necessitou de uma hora para responder todos os questionários, incluindo a sessão de treinamento. Todas as tarefas foram respondidas corretamente pelo participante. Vale destacar que o piloto ofereceu indícios de que a escolha pela

utilização de um modelo de processos para avaliação foi acertada, uma vez que o participante se sentiu familiarizado com a visualização e preferiu utilizá-la, mesmo quando outras visualizações respondiam a tarefa que estava sendo realizada de forma mais direta.

O estudo piloto indicou melhorias textuais para serem aplicadas nos instrumentos, com intuito de melhorar a clareza para os participantes acerca do que lhes era solicitado. Além disso, como algumas funcionalidades do ambiente não foram muito exploradas mesmo com a apresentação do material de treinamento, o piloto motivou a concessão de um tempo (cinco minutos) para que os participantes pudessem explorar o ambiente antes do início das tarefas, investigando as funcionalidades que os apoiariam para a realização das atividades.

Não foram observadas dificuldades na utilização do ambiente e/ou no entendimento das tarefas, não havendo qualquer impeditivo para a realização do estudo planejado.

## **6.4.2 Estudo Observacional**

Assim como no piloto, os participantes do estudo também foram selecionados por conveniência. No total, seis pessoas com diferentes níveis de formação – nas áreas de Engenharia de Software ou Sistemas de Informação – participaram do estudo. Não houve nenhum tipo de compensação para os participantes.

O procedimento também seguiu o mesmo definido para a execução do piloto. Foram realizadas sessões individuais com cada um dos participantes, sendo inicialmente entregues os formulários de consentimento e de caracterização. Em seguida, os participantes foram treinados utilizando o mesmo material, e tiveram a oportunidade de utilizar o ambiente EvoTrack-Process para explorar suas funcionalidades por um tempo máximo de 5 minutos. O limite de tempo não foi comunicado aos participantes, e apenas um precisou ser interrompido por usar o tempo completo. Assim como durante a execução do piloto, os participantes foram estimulados a informar o processo que utilizariam para responder as questões, indicando os passos que tomavam, enquanto um pesquisador responsável tomava notas e também observava a interação do participante com o ambiente.

O perfil dos participantes é apresentado com mais detalhes na próxima subseção.

### **6.4.2.1 Perfil dos Participantes**

Todos os participantes do estudo possuem pós-graduação completa ou em andamento, incluindo um com pós-doutorado completo, quatro doutorandos do

programa de pós-graduação em Engenharia de Sistemas e Computação da COPPE/UFRJ e um mestrando do programa de pós-graduação em Informática da UNIRIO (Figura 6.1).

A respeito do grau de experiência nas áreas de conhecimento relacionadas à esse trabalho, os participantes demonstraram características semelhantes: grande experiência nas áreas de processo, pouca experiência com a literatura acadêmica e práticas de colaboração, e alguma experiência com visualização de software. As exceções são de um participante que possui um grau de experiência bem maior em colaboração que os demais e outro na área de visualização de software (Figura 6.2).

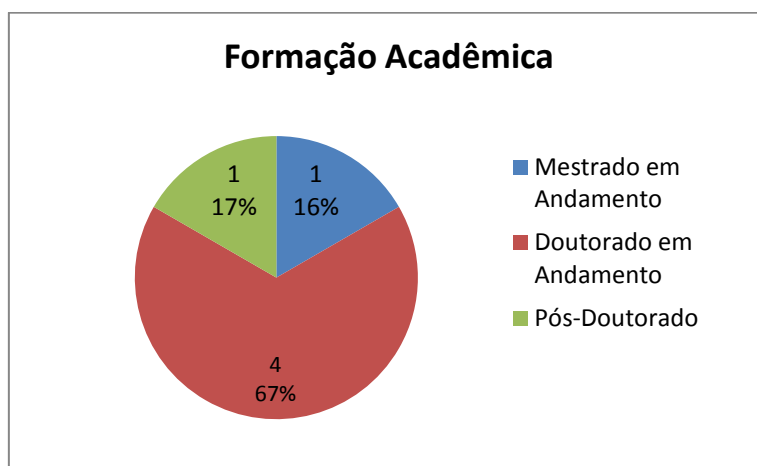


Figura 6.1. Formação acadêmica dos participantes do estudo.

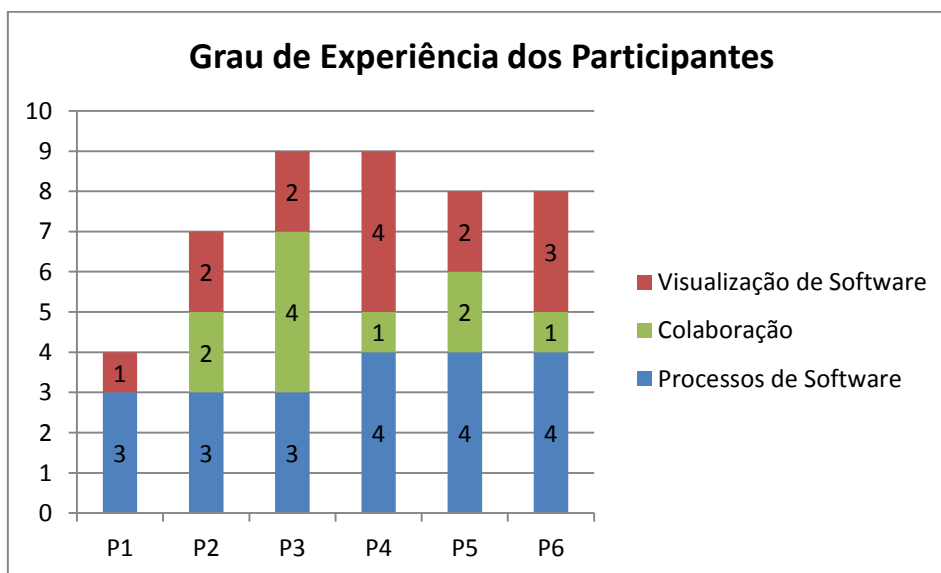


Figura 6.2. Grau de experiência dos participantes do estudo.

Os números no gráfico indicam o grau de experiência do participante em cada área de acordo com a seguinte escala: 0 = Nenhum; 1 = Estudado em aula ou em

livro; 2 = Praticado em projetos em sala de aula; 3 = Utilizado em projetos pessoais; 4 = Utilizado em projetos na indústria.

Embora em relação ao grau de experiência haja, em geral, pouca diferença entre os participantes, o mesmo não ocorre no que diz respeito ao tempo de experiência nos mesmos temas. O tempo de experiência com colaboração varia de 0 para um participante até 12 anos de um outro, com os demais variando entre alguns meses e 2 anos. Sobre a área de visualização de software, os valores também são bem variados mas com um desvio menor, com participantes em geral tendo alguns meses até dois anos de experiência e um único alcançando 5 anos. Finalmente, a área de processos de software é a que concentra o maior número de anos de experiência quando os valores são observados de forma agregada, também com valores variando muito de um participante para outro, de pouco menos de 1 ano até 10 anos (Figura 6.3).

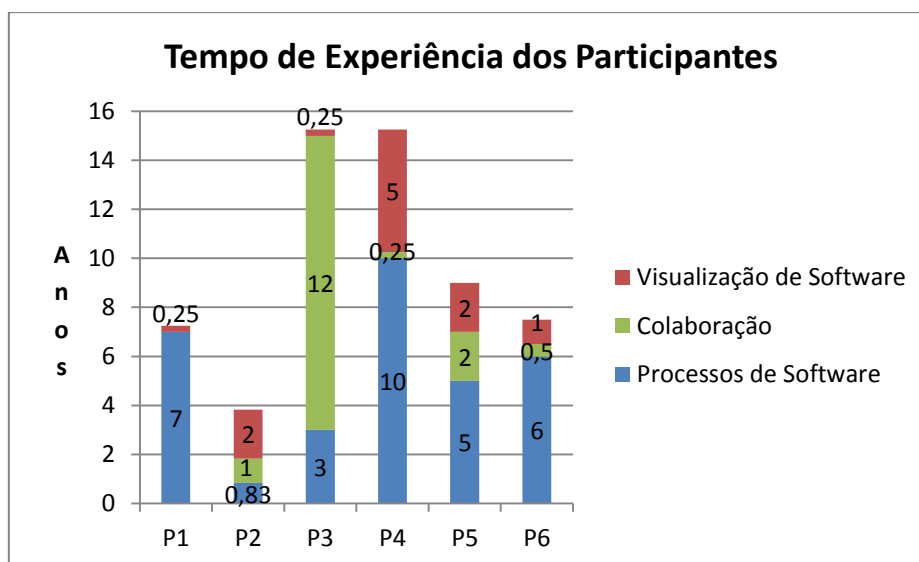


Figura 6.3. Tempo de experiência dos participantes do estudo.

Além do grau e tempo de experiência, os participantes indicaram seu grau de familiaridade com os domínios do contexto da avaliação: o desenvolvimento distribuído de software e, novamente, processos de software. A intenção é capturar como os participantes se posicionavam diante do tema, ou seja, se além de um determinado grau e tempo de experiência, os participantes também se viam como especialistas nos temas que iam observar ou não.

Outra informação importante é a experiência dos participantes com ambientes similares, isto é, com outras tecnologias e abordagens para acompanhamento de instâncias de processos. Nesse caso, embora a metade dos participantes tenham afirmado serem especialistas nesse tipo de tecnologia – independentemente de terem

ou não tido experiência na indústria – outros dois disseram não ter qualquer familiaridade com o domínio, incluindo um com grau considerável de experiência no tema processos de software.

Os valores indicados pelos participantes foram sumarizados nos dois gráficos abaixo (Figura 6.4 e Figura 6.5).

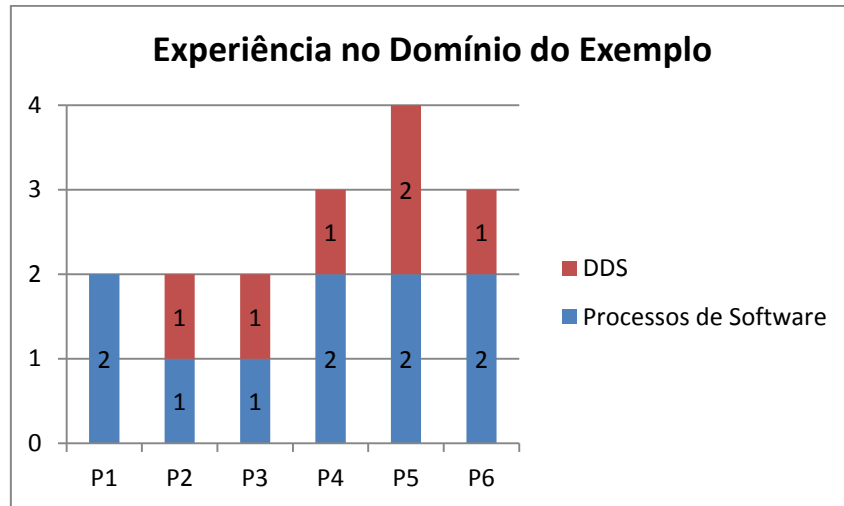


Figura 6.4. Grau de experiência dos participantes do estudo nas áreas do domínio observado.

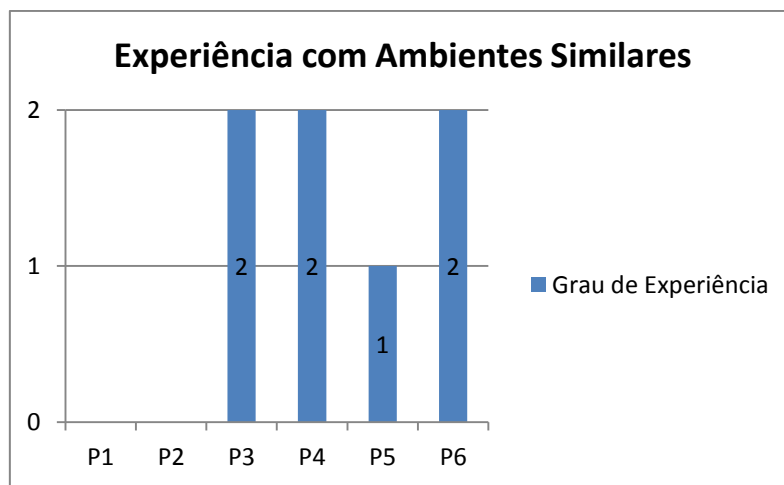


Figura 6.5. Experiência dos participantes com ambientes de acompanhamento de processos.

Os números no gráfico indicam o grau de experiência do participante em cada área de acordo com a seguinte escala: 0 = Nenhuma familiaridade; 1 = Alguma familiaridade; 2 = Especialista no domínio.

#### 6.4.2.2 Análise do Desempenho das Tarefas

Todos os participantes apresentaram respostas corretas para as tarefas de filtragem, e, portanto, todos os participantes foram considerados aptos para participar

do estudo, não havendo descarte de dados. Em relação ao desempenho de realização das tarefas básicas, os participantes acertaram em aproximadamente 80% dos casos, totalizando em 4 acertos para tarefa 2, 5 acertos para a tarefa 3 e todas as 6 respostas corretas para a tarefa 5. Sobre a única Tarefa de Assimilação – tarefa 6 – apenas um participante ofereceu resposta satisfatória para a questão.

Os gráficos abaixo (Figuras 6.6 e 6.7) resumizam o desempenho geral dos participantes: o primeiro destaca o número de acertos de cada tarefa; enquanto o segundo mostra a porcentagem de tarefas corretas por tipo, com a contribuição de cada tarefa para a porcentagem. Em seguida, cada uma das tarefas e suas respectivas observações são discutidas individualmente.

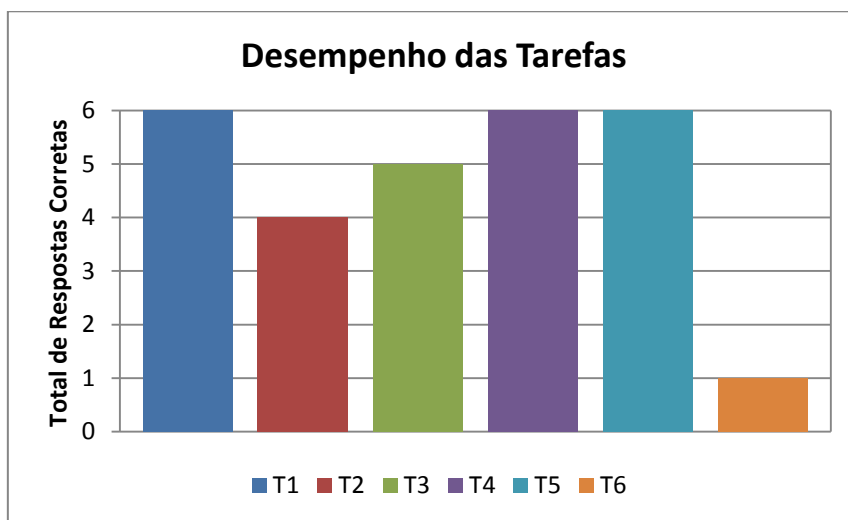


Figura 6.6. Acertos dos participantes do estudo na execução das tarefas.

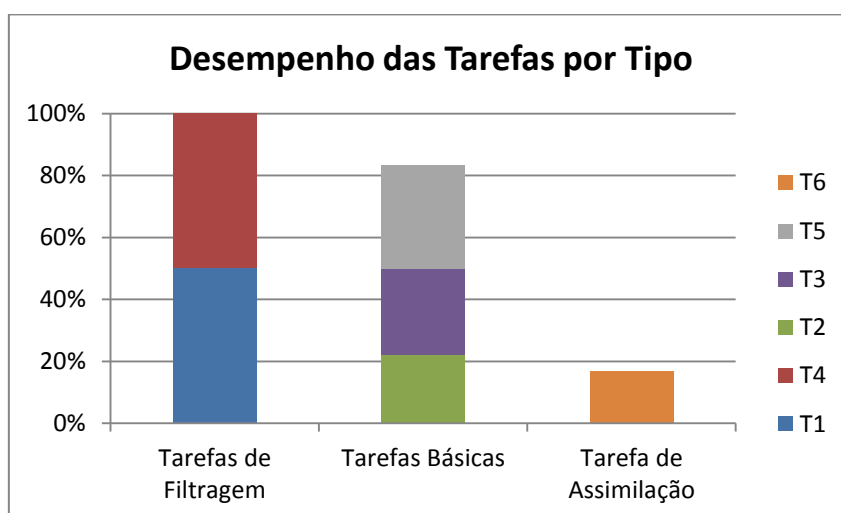


Figura 6.7. Acertos dos participantes por tipo de tarefa.



## 6.5 Análise dos Resultados

A análise dos resultados do estudo foi feita considerando: i) as respostas dadas pelos participantes para cada uma das tarefas e notas de utilização do ambiente tomadas pelo pesquisador responsável durante cada sessão; e ii) as respostas dos participantes para as questões do questionário de avaliação do estudo. Os dados coletados não foram codificados ou agregados seguindo alguma técnica específica, sendo utilizadas apenas as interpretações e impressões do pesquisador responsável para deduzir as conclusões apresentadas a seguir.

### 6.5.1.1 Tarefa 1

A tarefa 1 consistia na identificação de todos os atores que participaram de duas instâncias do processo. A tarefa era considerada de filtragem, tendo em vista que o discernimento de todos os envolvidos durante a execução de um processo é imprescindível para entender e, eventualmente, acompanhar como a colaboração ocorreu na empresa. Todas as visualizações oferecidas pelo ambiente são capazes de responder a questão, cada uma requisitando mais ou menos interações por parte de usuários: ambas as visualizações do modelo do processo e rede social exigiram recursos de *zoom* e deslocamento (*pan*), para que o participante conseguisse efetivamente identificar todos os participantes; enquanto a tabela de métricas oferecia a informação de forma direta, sem valores duplicados.

Mesmo com todas as visualizações sendo apresentadas durante o treinamento e também disponibilizadas para a utilização dos participantes durante os 5 minutos para se habituarem com o ambiente, as notas de todas as sessões referentes a essa tarefa indicaram a utilização do modelo de processos para a execução da atividade. Dessa forma, os participantes lidaram ainda com a dificuldade adicional imposta pela visualização, que repete os ícones dos atores toda vez que estes participam de uma atividade diferente como recurso para minimizar a quantidade de arestas se sobrepondo a outros elementos do modelo.

Esse comportamento por parte de todos os participantes reforçou o argumento utilizado para o desenvolvimento da visualização do modelo, fundamentada na ideia de que é uma representação comum para pessoas com familiaridade no domínio de processos de software e, conseqüentemente, de mais fácil aceitação do que uma nova metáfora visual. Dessa forma, mesmo não sendo a visualização que mostrava os dados de forma mais direta, esse era o comportamento esperado pelos participantes do estudo.

Entre as eventuais dificuldades encontradas para responder a tarefa, os participantes tiveram que usar os mecanismos de interação da visualização (*pan*, *zoom*, *zoom-to-fit*) para ter uma visão detalhada do processo por inteiro, uma vez que o modelo de processos representado ocupava um espaço maior do que o disponível na tela das sessões do estudo. No entanto, as notas tomadas durante as observações dos participantes indicam que não houve qualquer dificuldade de utilização desses recursos interativos após um período de alguns segundos de adaptação.

### **6.5.1.2 Tarefa 2**

A tarefa 2 consistia em identificar com quais outros atores de sítios diferentes um determinado ator colaborou em uma determinada instância de processo. A tarefa foi planejada com o objetivo de permitir que os participantes interagissem com a visualização de rede social, possibilitando eventuais comentários e contribuições para a visualização e reforçando a própria existência da visualização para que pudesse ser utilizada nas próximas tarefas. A tarefa também visava investigar se o ambiente EvolTrack-Process oferece facilidades para evidenciar os relacionamentos inter/intra-sítios de desenvolvimento da organização, se baseando na revisão da literatura da área de desenvolvimento distribuído que alega que a distância é um fator impactante na colaboração.

O ator mencionado na questão colaborou (i.e., executou outra atividade com) apenas um outro participante em todo o processo observado – que efetivamente estava em outro sítio – e os participantes que procurassem por essa informação utilizando o modelo de processos teriam facilidade para obter a resposta. Apenas um participante não usou a visualização do modelo do processo para verificar se o ator efetivamente participou da instância indicada na questão.

Todos os participantes utilizaram a visualização de rede social como base para responder a questão, e foi observado que todos eles tiveram dificuldades para acompanhar o relacionamento do participante, uma vez que a distância entre os sítios era longa. Essa dificuldade percebida por todos os participantes oferece indícios de que os mecanismos de interação pensados para a rede social (*zoom* e *pan*) não são suficientes para destacar os dados em redes sociais com apoio do mapa quando os nós estão distribuídos em localidades muito distantes. A possibilidade de uma rede social comum, sem o mapa como fundo, mas com um indicativo da localidade de cada participante pode ser uma solução para o problema identificado.

Também é importante mencionar que nenhum participante se deu conta de que uma consulta rápida da rede social era suficiente para responder a questão: como o ator só interagiu com uma outra pessoa que estava em outro sítio, bastava constatar

que a aresta realmente se destinava para outra localidade na rede social e buscar pelo relacionamento do ator indicado na questão usando o modelo do processo. A construção do ambiente vislumbrou que todas as visualizações juntas em uma mesma tela poderia estimular a combinação de várias fontes diferentes de informação para obter um entendimento e reconhecimento de características da colaboração em processos de forma mais simples e rápida, mas as observações de utilização do ambiente para responder essas e outras tarefas dão indícios de que mais mecanismos que interliguem essas visualizações devam ser necessários, indo além da apresentação de múltiplas visualizações integradas em uma mesma tela.

### **6.5.1.3 Tarefa 3**

A tarefa 3 consistia em identificar todos os atores que já participaram da execução de uma determinada atividade em conjunto com um determinado ator, em todas as instâncias de processo observadas. Apenas a visualização do modelo de processos poderia ser capaz de auxiliar os participantes na elaboração da resposta, com filtros de visibilidade auxiliando na eliminação de todas as outras atividades da visualização que não aquela específica exigida pela tarefa. Essa tarefa complementa a Tarefa 1 no contexto de entendimento da colaboração, investigando se além de identificar os envolvidos na execução das instâncias, o ambiente também auxilia a entender os relacionamentos entre essas pessoas, ou seja, com quem cada uma delas colaborou durante a execução das instâncias de processo.

Embora a visualização do modelo de processos mostre os atores que participaram de uma atividade de forma direta, isto é, sem a necessidade de interação ou sujeito à interpretação, não há no ambiente uma forma para considerar essa informação considerando todas as instâncias do processo agregadas, exigência para responder corretamente a tarefa. Essa limitação da ferramenta foi considerada durante o planejamento da tarefa, proposta com objetivo de coletar indícios sobre outras metáforas visuais ou mecanismos de interação que os participantes do estudo pudessem oferecer.

Os participantes do estudo consideraram as funcionalidades do ambiente por um tempo maior antes de indicar como esperavam responder a questão, alegando estarem pensando em como considerar mais de uma instância ao mesmo tempo. Após reconhecerem que não havia mecanismos que facilitassem a análise, todos os participantes recorreram imediatamente ao modelo do processo e navegaram pelas instâncias, com metade deles dizendo que a atividade era fácil embora trabalhosa, provavelmente porque embora não fosse necessário raciocinar muito para encontrar as informações para responder a questão usando as visualizações, os participantes

ainda precisavam navegar por várias instâncias, gerando um trabalho de certa forma repetitivo.

Em relação a aplicabilidade das técnicas de visualização utilizadas, dois participantes recomendaram espontaneamente durante a execução da tarefa que o modelo de processos deveria ser capaz de concentrar várias instâncias para auxiliar nesses casos, com cores ou outro tipo de indicador visual empregado para diferenciar quais elementos pertenciam a qual instância. Nenhum participante sugeriu a utilização de outra metáfora visual ou técnica de visualização.

#### **6.5.1.4 Tarefa 4**

A tarefa 4 consistia em identificar qual ator colaborou com o maior número de outros atores em uma determinada instância do processo. Essa tarefa foi posicionada nessa ordem para que os participantes não fossem levados a utilizar a visualização da tabela de métricas para resolver as tarefas em que ela poderia ser mais apropriada sem que buscassem pela visualização espontaneamente, como no caso da tarefa 1. O objetivo da tarefa era verificar se os participantes do estudo conseguem identificar facilmente os atores mais importantes de um processo segundo algum critério (no caso da tarefa, o número de relacionamentos).

A resposta para a tarefa podia ser obtida utilizando qualquer uma das visualizações oferecidas pelo ambiente, das quais a tabela de métricas era a única que oferecia a resposta de forma direta. Todos os participantes utilizaram a tabela para responder a tarefa, com dois deles usando a medida de “grau de centralidade” para concluir a conectividade dos atores enquanto os outros quatro utilizaram o número de conexões. Dois participantes cogitaram ainda utilizar alguma das outras medidas, mas alegaram desistir por acharem que outra exprimia melhor a informação necessária para responder a tarefa.

Não foram reportadas ou percebidas quaisquer dificuldades para observar as propriedades de rede sociais disponibilizadas pelo ambiente, não sendo levantada a necessidade de nenhum outro mecanismo de interação para a visualização além do único já implementado (ordenação de valores).

#### **6.5.1.5 Tarefa 5**

A tarefa 5 consistia em identificar quais atividades diferentes um determinado ator desempenhou em duas instâncias do processo. A tarefa é muito similar à tarefa 3, com o ambiente apresentando a informação de forma direta, porém sem recursos para comparar duas instâncias diretamente. A tarefa também tem por objetivo investigar as facilidades oferecidas pelo ambiente para o acompanhamento dos relacionamentos de colaboração de um determinado ator.

Durante a execução dessa tarefa os participantes hesitaram muito menos que durante a execução da tarefa 3, quando ainda precisaram explorar o ambiente para ter certeza de que não havia uma forma mais fácil de alcançar os dados que precisavam. Dois dos participantes reforçaram nesse momento a ideia de ampliar a visualização do modelo de processos para considerar múltiplas instâncias, e novamente o uso de cores ou ícones diferentes para os elementos de diferentes instâncias foram sugeridos. Assim como na tarefa 3, nenhum participante sugeriu a criação de um outro mecanismo, indicando ser suficiente o aumento das capacidades da visualização do modelo utilizado.

#### **6.5.1.6 Tarefa 6**

A tarefa 6 consistia em substituir um determinado ator na realização de uma atividade de uma instância. A tarefa é de certa forma subjetiva, tendo em vista que cada organização e, conseqüentemente, cada participante pode adotar seus próprios critérios para escolher qual ator está mais apto para a realização de uma atividade no lugar de outro. A tarefa foi concebida para ver quais insumos o ambiente conseguiria fornecer para os usuários para tarefas que envolvessem raciocínio, investigando a capacidade da ferramenta no auxílio à tomada de decisão.

Diante dessa subjetividade, foram estabelecidos critérios objetivos para considerar a resposta de um participante correta, sendo necessário que o participante contemplasse pelo menos duas informações disponibilizadas pelo ambiente: os outros atores que participaram daquela tarefa e os sítios onde eles estavam localizados.

Apenas um participante do estudo se preocupou em verificar a localidade geográfica dos atores para a alocação do substituto, uma informação discutida amplamente no contexto de trabalhos sobre o desenvolvimento distribuído de software (DDS). Embora o tamanho da amostra de participantes não seja grande e um deles inclusive alegou não ter qualquer experiência no domínio de DDS, a informação das localidades foi amplamente descartada, com um participante inclusive diminuindo a área das demais visualizações para ampliar o modelo de processo exibido.

O critério para escolha da resposta foi sempre considerando qual outro ator já havia desempenhado a tarefa mais vezes em instâncias anteriores. Três participantes também notaram os relacionamentos do participante substituído, buscando encontrar outros atores que já houvessem colaborado na mesma tarefa antes. Um participante ainda optou por um determinado ator porque ele participava da instância do processo observada, alegando acreditar que por essa razão o ator deveria estar disponível para a execução.

Do ponto de vista do objetivo do estudo, essa tarefa novamente reafirmou que os participantes tiveram pouco interesse em utilizar as múltiplas perspectivas oferecidas, se contentando apenas com as informações dispostas no modelo do processo. O fato da visualização de rede social ter sido ignorada em várias tarefas do estudo leva a pensar que a forma como a visualização foi construída não foi interessante para os participantes, podendo haver maneiras mais simples e diretas para representar a distância entre os nós que a visão do mapa terrestre adotada.

### **6.5.2 Análise do Questionário de Avaliação**

Com base nas respostas do questionário de avaliação do estudo entregue aos participantes (Apêndice B), foi realizada uma análise qualitativa complementar da percepção dos participantes, incluindo questões sobre as próprias tarefas e sobre o apoio do ambiente EvoTrack-Process na realização de cada uma das tarefas, bem como das técnicas e mecanismos disponibilizados no ambiente.

Sobre a capacidade de realização das atividades, considerando as três respostas possíveis – consegui, consegui parcialmente e não consegui – todos os participantes do estudo julgaram que conseguiram realizar as atividades propostas. Um participante acrescentou à sua resposta a sugestão de criação de um mecanismo de interação para exibir todos os atores que já participaram de uma determinada atividade do processo, concedendo à visualização a capacidade de exibir um conjunto agregado de instâncias especificado pelo usuário.

Em relação à satisfação com suas respostas para as tarefas, um único participante informou estar apenas parcialmente satisfeito em uma escala que considerava três respostas possíveis: satisfeito, parcialmente satisfeito e insatisfeito. Todos os demais se mostraram satisfeitos. Em uma justificativa de sua resposta, o participante parcialmente satisfeito alegou que não sentia confiança na resposta dada para a tarefa 6, pois gostaria de contar com outros indicadores do trabalho dos envolvidos no processo, citando como exemplo dados sobre a entrega das atividades dos atores (e.g., se um determinado ator entregou a atividade no prazo esperado ou não).

Analisando a possibilidade de visualização da colaboração com as informações apresentadas, os participantes se mostraram divididos: metade deles acharam que era possível, enquanto a outra metade afirmou ser apenas parcialmente possível. No espaço para comentários, dois participantes alegaram não ser possível afirmar com certeza se as pessoas colaboraram ou como se deu essa colaboração utilizando apenas o ambiente. Entretanto, essa limitação levantada pelos participantes faz parte da natureza de tecnologias que se baseiam em dados coletados a partir de

repositórios de informações, que são limitados a apresentar o que aconteceu e não como aconteceu ou as decisões por trás das ações. Nenhum participante sinalizou achar impossível observar a colaboração com as informações apresentadas.

Quando solicitados para responder sobre a dificuldade das tarefas do estudo, os participantes alegaram que elas eram fáceis, com apenas um indicando que eram muito fáceis. Eles também afirmaram que o ambiente EvoTrack-Process facilitou bastante para responder as tarefas, com novamente apenas um participante discordando e indicando que o ambiente facilitou um pouco. Nenhum participante escolheu as demais opções de resposta, sendo elas: indiferente; dificultou um pouco; dificultou bastante. No espaço destinado para comentários dessas perguntas, três participantes elogiaram a facilidade de utilização do ambiente, mas se queixaram da impossibilidade de ver mais de uma instância de processo ao mesmo tempo.

A respeito da facilidade que as técnicas de visualização disponibilizadas pelo ambiente proporcionaram para a realização das tarefas, quatro dos participantes indicaram que elas facilitaram bastante, enquanto outros dois disseram ter facilitado um pouco. Nenhum participante optou pelas demais respostas, que obedeceram a mesma escala da questão anterior: indiferente; dificultou um pouco; e dificultou bastante. Os participantes não justificaram a escolha das respostas.

Sobre as funcionalidades, aspectos positivos e negativos do ambiente, os participantes elogiaram os filtros e mecanismos de interação disponíveis para as visualizações, com dois participantes destacando os dados exibidos por serem mais completos que outras ferramentas de acompanhamento de processos. A disposição e navegação entre as instâncias para permitir entender como ocorreu a execução do processo também foi enaltecida, assim como a visão de fluxos mais comuns de processos, que chamou atenção de um participante pela originalidade e utilidade do mecanismo. Os aspectos negativos ficaram por conta de alguns problemas de desempenho e da manipulação dos filtros, que em vários momentos exigiam que o usuário tomasse passos adicionais para que eles fossem efetivamente aplicados às visualizações. A tabela de métricas foi tida como a visualização menos relevante para as tarefas desempenhadas. Um participante também demonstrou arrependimento por não ter considerado as informações geográficas para as tarefas quando contemplou sobre os pontos positivos e negativos do ambiente, e a visualização da rede social georreferenciada foi mencionada duas vezes entre os aspectos positivos, mesmo quase não tendo sido utilizada nas sessões.

Em suas considerações finais, os participantes ofereceram várias sugestões para usabilidade da ferramenta e fizeram alguns elogios e críticas adicionais. Poucas dessas sugestões tinham relação com as visualizações do ambiente, sendo estas

últimas sempre relacionadas a necessidade de observar várias instâncias simultaneamente nas visualizações.

## **6.6 Considerações Finais**

Este capítulo apresentou a avaliação do protótipo EvolTrack-Process através de um estudo observacional. O objetivo do estudo era investigar se as visualizações projetadas para o ambiente EvolTrack-Process seriam capazes de auxiliar os seus usuários a compreender e acompanhar como ocorreu a colaboração em instâncias de processos de software. Cada participante executou tarefas predeterminadas utilizando o ambiente, sendo estimulados a comunicar suas ações enquanto um pesquisador tomava notas a partir dos comentários e de suas próprias observações sobre as interações dos participantes com o ambiente.

As observações dos participantes utilizando o ambiente possibilitou a identificação de várias limitações e oportunidades de melhoria para o ambiente. No que diz respeito a investigação das técnicas de visualização e mecanismos de interação, o modelo de processos e seus filtros foram os mais utilizados, com nenhum participante demonstrando dificuldade para utilizar a visualização. A visualização de rede social georreferenciada sobre o mapa foi elogiada por alguns participantes, mas foi pouco utilizada durante a execução das tarefas e foram tomadas algumas notas que indicam dificuldades de utilização, motivando a criação de uma metáfora visual ou mecanismos de interação diferentes para exibir as informações. Finalmente, a tabela de métricas foi pouco explorada durante as tarefas, mas não foram identificadas dificuldades ou construção de novos mecanismos, aparentemente servindo o seu propósito de apoiar a interpretação da colaboração por meio da análise de propriedades de redes sociais.

É importante notar que os participantes do estudo não eram especialistas no domínio de Visualização de Software, o que pode ter limitado as contribuições dos participantes nesse sentido. Entretanto, a população do estudo se aproxima dos potenciais usuários do ambiente, que também podem não ter um vasto conhecimento de técnicas de visualizações, e os indícios encontrados a partir das observações do comportamento dos participantes – o que buscavam por primeiro e como interagiam com o ambiente – podem ser valiosos para a construção de outras soluções que planejem se utilizar dessas técnicas no contexto de processos.



## 7 Considerações Finais e Trabalhos Futuros

*Neste capítulo, as considerações finais desta dissertação são apresentadas, resumindo sua motivação e proposta, destacando contribuições e limitações. Ao final, são descritos um conjunto de trabalhos futuros planejados para dar continuidade à pesquisa aqui apresentada.*

### 7.1 Epílogo

O desenvolvimento de software é uma atividade intensamente colaborativa, onde várias pessoas assumindo diferentes papéis precisam se coordenar e interagir umas com as outras. Essa característica torna a colaboração essencial para o desenvolvimento de sistemas de software, e, através da colaboração, atores podem complementar suas competências e esforços individuais, trocar conhecimentos, ampliar a capacidade de tomar decisões e construir um ambiente de trabalho de melhor convivência e de maior produtividade.

No entanto, alcançar os benefícios associados à colaboração não é trivial, e há uma série de fatores que podem tornar o alcance dessas vantagens um desafio, desde problemas de relacionamento e de ambientes de trabalho até falta de planejamento, desconhecimento de competências e alocação de times formados por membros que trabalham pior juntos do que outros. Além disso, no contexto moderno de desenvolvimento de software, onde empresas estão buscando o estabelecimento de vários sítios de desenvolvimento para obter vantagens competitivas, surgem ainda mais complicações – barreiras geográficas, temporais e culturais – que dificultam a capacidade dos atores no trabalho em conjunto durante a construção do software.

Essa dissertação investigou mecanismos para apoiar gerentes de processos de desenvolvimento de software no entendimento e acompanhamento da colaboração em seus processos. A solução proposta no trabalho parte da premissa de que técnicas de visualização de software poderiam facilitar esse entendimento e acompanhamento, oferecendo uma visão geral das instâncias de processo já finalizadas ou ainda em execução em conjunto com informações de quais atores desempenharam cada atividade, onde estavam, e quantificando esses relacionamentos em termos de propriedades de análises de redes sociais. A partir da utilização dessas técnicas, era esperado construir uma imagem geral de como se deu a execução de uma instância de software, facilitando a interpretação dos dados e reduzindo a sobrecarga de

informações quando comparando a utilização dessas técnicas com a análise de *logs* de execução de processos.

As revisões da literatura que embasaram esse estudo incluíram a própria área de colaboração em processos de software (Capítulo 2) e a de visualização de software (Visualização de Software 3). Mesmo com a colaboração sendo uma característica de grande importância para o desenvolvimento de software, não foram encontradas alternativas que tivessem o foco adotado pelo ambiente criado nessa dissertação: a construção de mecanismos visuais para apoio do entendimento da colaboração a partir de dados de processos de software.

A partir do reconhecimento dessa lacuna, foram identificadas algumas características e requisitos para auxiliar a entender a colaboração em processos de software, incluindo técnicas de visualização de software e de avaliação da interação entre atores por meio de propriedades de redes sociais, entre outras.

O ambiente EvoTrack-Process, apresentado nessa dissertação, incorpora as características propostas, disponibilizando três visualizações e alguns mecanismos de interação para lidar com o problema descrito anteriormente. As visualizações oferecidas pelo ambiente foram foco de um estudo observacional, planejado para avaliar indícios da utilização das técnicas adotadas para representação da colaboração em processos de software. O estudo aponta para indicadores positivos, sobretudo em relação à utilização de modelos de processo para apresentação de informações dessa natureza.

Esse trabalho também está inserido no contexto de uma iniciativa maior que tem por objetivo explorar aspectos colaborativos em processos de software. Nesse sentido, o trabalho se apresenta como um mecanismo novo à sistemática COMPOOTIM, complementando a sistemática com dados oriundos de *logs* de execução de processo e apresentando técnicas de visualizações novas para oferecer outra perspectiva da colaboração em processos. Além disso, o trabalho une duas áreas de pesquisa atualmente investigadas pelo grupo de pesquisa de Reutilização de Software da COPPE/UFRJ, sendo estas as áreas de processos e de visualização de software.

O restante desse capítulo está organizado como segue: a Seção 7.2. sumariza as contribuições do trabalho; enquanto a Seção 7.3. discute as limitações percebidas a partir de uma análise crítica; finalmente, a seção 7.4. apresenta as oportunidades e ideias levantadas para dar continuidade ao trabalho descrito nessa dissertação.

## **7.2 Contribuições**

As contribuições desse trabalho são apresentadas em duas perspectivas distintas: tecnológica e científica.

### **7.2.1 Contribuições Tecnológicas**

Do ponto de vista tecnológico, a principal contribuição dessa proposta é o próprio ambiente projetado, denominado EvoTrack-Process. Nesta dissertação, foi apresentado o projeto técnico e detalhes de desenvolvimento de um protótipo do ambiente, voltado para auxiliar o entendimento e acompanhamento da a colaboração em processos de software.

Em conjunto com a definição do ambiente e desenvolvimento do protótipo, é possível sumarizar outras duas contribuições tecnológicas dessa dissertação:

- Identificação de características relevantes para abordagens dedicadas a facilitar o entendimento e acompanhamento da colaboração em processos de software;
- Expansão do ambiente EvoTrack e da etapa de monitoração da sistemática COMPOOTIM, apresentando um conjunto de soluções tecnológicas novas para estudar o desenvolvimento de software.

### **7.2.2 Contribuições Científicas**

Em relação à perspectiva científica, a contribuição mais importante desse trabalho está na avaliação do uso de visualizações para acompanhar a colaboração em processos de software. Através do estudo realizado, foi possível coletar indícios sobre a utilização das visualizações adotadas nessa proposta. Os resultados do estudo indicam que os mecanismos de visualização são válidos para apoio em tarefas associadas ao entendimento e acompanhamento da colaboração em processos de software.

Além dos indícios encontrados pelo estudo de avaliação, é possível sumarizar outras duas contribuições científicas desse trabalho:

- Realização de uma revisão bibliográfica sobre colaboração e mecanismos de visualização para ressaltá-la no desenvolvimento de software
- Elaboração de um pacote (incluindo o plano e demais instrumentos) para possíveis replicações do estudo executado com outras populações, aumentando o grau de confiança nos indícios encontrados. O pacote também pode servir também como base para estudos em contextos similares, auxiliando no planejamento de outros estudos futuros.

## 7.3 Limitações

Algumas limitações desse trabalho foram identificadas através de uma análise crítica do ambiente e dos resultados do estudo realizado. Em particular, destacam-se as quatro limitações listadas a seguir:

- Embora o conjunto de informações utilizado na abordagem reflita algumas características de colaboração em processos, ele não se baseou em ontologias ou foi definido através de uma análise completa dos dados que melhor refletiriam esse aspecto, sendo fundamentado em outros trabalhos que também coletam dados de *logs* de ambientes de execução de processos. Nesse sentido, o conjunto de informações pode ser visto como uma limitação do trabalho, ainda que se acredite que foram selecionados os dados com maior potencial para oferecer entendimento da colaboração em processos;
- O ambiente construído se beneficiou da arquitetura projetada pelo ambiente EvoTrack, reutilizando soluções que foram importantes para o desenvolvimento de vários módulos. No entanto, essa reutilização também trouxe uma limitação estrutural: o ambiente funciona apenas vinculado à IDE Eclipse, sob a forma de um *plug-in*. Além disso, o EvoTrack-Process é capaz de coletar informações de processo apenas do BPMS Bonita;
- As visualizações são um grande diferencial desse trabalho, sob a premissa de que elas podem auxiliar no acompanhamento da colaboração em processos. Embora o estudo de observação realizado ofereça indícios de que a hipótese é verdadeira, o estudo contou com um número limitado de visualizações, e outras técnicas podem produzir resultados diferentes. Outra limitação diz respeito à escalabilidade das visualizações, onde apesar de terem sido disponibilizados filtros e mecanismos de interação pensados para lidar com a sobrecarga de informações, não há garantias de que as visualizações utilizadas são capazes de suportar o entendimento da colaboração com a mesma qualidade em processos com um nível de complexidade muito diferente dos observados nessa dissertação;
- A avaliação do ambiente foi limitada, tendo sido realizado apenas um estudo observacional para identificar o apoio das visualizações para o entendimento e acompanhamento da colaboração em processos. Para que seja possível obter evidências mais significativas sobre os benefícios reais que o ambiente é capaz de oferecer, se faz necessário o planejamento e condução de experimentos adicionais, com um número maior de participantes.

## 7.4 Trabalhos Futuros

Foram identificadas algumas possibilidades de extensão da pesquisa apresentada nesse trabalho, tanto sob um ponto de vista ferramental das características planejadas para o ambiente como do ponto de vista da avaliação realizada. Entre essas possibilidades identificadas, destacam-se:

- A ampliação do estudo realizado, usando como fonte de dados *logs* de processos mais complexos e maiores. Dessa forma, será possível investigar a capacidade de extensibilidade das visualizações, bem como levantar novas melhorias necessárias para o ambiente;
- A replicação de estudos com outras populações, reforçando ou refutando indícios encontrados e, se possível, realizar análises com fundamentação estatística;
- Incorporação do ambiente junto com os demais mecanismos da COMPOOTIM, dando origem a uma interface integrada com outros mecanismos desenvolvidos na sistemática;
- Evolução e manutenção dos mecanismos disponíveis no ambiente, incluindo novas fontes de dados de processo, novos mecanismos de interação e novos recursos visuais para expandir a capacidade do EvolTrack-Process. A própria avaliação executada nesse trabalho ofereceu diversos pontos de melhoria para as visualizações (e.g. a visualização com o mapa) e mecanismos de interação, servindo como base para versões futuras.

Essas extensões representam passos importantes para expandir o conhecimento sobre a colaboração no desenvolvimento de software, sobretudo com o viés de processos de software. Através da consideração dos dados de instâncias de processos, comumente ignorados em estudos relacionados, é esperado oferecer uma perspectiva nova para o entendimento e acompanhamento da colaboração, ampliando a visão dessa característica tão importante para a Engenharia de Software.

## Referências Bibliográficas

- ÅGERFALK, P. et al. Benefits of Global Software Development: The Known and Unknown. In: WANG, Q.; PFAHL, D.; RAFFO, D. M. **Making Globally Distributed Software a Success Story, ICSP 2008**. [S.l.]: Springer-Verlag Berlin Heidelberg, 2008. p. 1-9.
- AGILE ALLIANCE. **Manifesto for Agile Software Development**. [S.l.]. 2001.
- ALCANTARA, J. **La neutralidad de la Red**. [S.l.]: El Arte de las Cosas, 2010.
- ARAUJO, R. M.; BORGES, M. R. S. The role of collaborative support to promote participation and commitment in software development teams. **Software Process: Improvement and Practice**, 12, n. 3, 2007. 229-246.
- ARBAOUI, S. et al. A Comparative Review of Process-Centered Software Engineering Environments. **Annals of Software Engineering**, v. 14, n. 1-4, p. 311-340, 2014.
- ARENT, J.; NORBJERG, J. **Software Process Improvement as Organizational Knowledge Creation: A Multiple Case Analysis**. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS '00). Hawaii, USA: Computer Society Press. 2000. p. 1-11.
- BALIEIRO, M. A.; SOUSA JÚNIOR, S. F.; DE SOUZA, C. R. B. **Facilitating Social Network Studies of FLOSS using the OSSNetwork**. Proceedings of the International Conference on Open Source Systems (OSS'08). Milano, Itália: Springer. 2008. p. 343-350.
- BALL, T. et al. **If your version control system could talk**. ICSE Workshop on Process Modelling and Empirical Studies of Software Evolution. Boston: [s.n.]. 1997.
- BALOIAN, N. et al. **A collaborative mobile approach for business process elicitation**. Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD). Lausanne, Switzerland: IEEE. 2011. p. 473-480.
- BANNON, L. J.; KUUTTI, K. **Shifting perspectives on organizational memory: from storage to active remembering**. Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences (HICSS'96). Hawaii, USA: Computer Society Press. 1996. p. 156-167.
- BARABASI, A. L.; BONABEAU, E. Scale-Free Networks. **Scientific**, p. 50-59, 2003.

- BARNES, S.; GOLDENA, B.; WASILB, E. Exploring the effects of network structure and healthcare worker behavior on the transmission of hospital-acquired infections. **IIE Transactions on Healthcare Systems Engineering**, v. 2, n. 4, Dezembro 2012.
- BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The Goal Question Metric Approach. **Encyclopedia of Software Engineering**, 1994.
- BEGEL, A.; HERBSLEB, J. D.; STOREY, M. **The future of collaborative software development**. Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion (CSCW '12). New York: ACM. 2012. p. 17-18.
- BENDRAOU, R. et al. A Comparison of Six UML-Based Languages for Software Process Modeling. **IEEE Transactions on Software Engineering**, v. 36, n. 5, p. 662-675, Setembro-Outubro 2010. ISSN 0098-5589.
- BIEHL, J. et al. **FASTDash**: A visual dashboard for fostering awareness in software teams. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07). San Jose, USA: ACM. 2007. p. 1313-1322.
- BRAMBILLA, M.; FRATERNALI, P.; RUIZ, C. K. V. **Combining Social Web and BPM for Improving Enterprise Performances**: the BPM4People Approach to Social BPM. Proceedings of the 21st international conference companion on World Wide Web (WWW '12 Companion). New York: ACM. 2012. p. 223-226.
- BROOKS, F. P. **The Mythical Man-Month**: Essays on Software Engineering. 1st. ed. Boston, USA: Addison-Wesley Longman Publishing Co., 1978.
- BROWN, R. A.; RECKER, J. C.; WEST, S. Using virtual worlds for collaborative business process modeling. **Journal of Business Process Management**, v. 17, n. 3, p. 546-564, Maio 2011. ISSN 1463-7154.
- CARD, S.; MACKINLAY, J.; SHNEIDERMAN, B. **Readings in Information Visualization**: Using Vision to Think. 1. ed. [S.I.]: Morgan Kaufmann Publishers, 1999.
- CATALDO, M.; HERBSLEB, J. D. **Architecting in software ecosystems**: interface translucence as an enabler for scalable collaboration. Proceedings of the Fourth European Conference on Software Architecture (ECSA '10). [S.I.]: ACM. 2010. p. 65-72.
- CHEN, C. **Information Visualization**: Beyond the Horizon. 2. ed. [S.I.]: Springer, 2006.

- CLARET, M. D. **Métricas para Colaboração em Processos de Negócio**. Workshop de Teses e Dissertações - Simpósio Brasileiro de Sistemas de Informação (SBSI). João Pessoa, Brasil: [s.n.]. 2013. p. 31-36.
- CONWAY, M. E. How do committees invent. **Datamation**, 14, n. 4, 1968. 28-31.
- COSTA, G. C. B. et al. **Monitoring Collaboration in Software Processes Using Social Networks**. Proceedings of the 20th International Conference on Collaboration and Technology (CRIWG 2014). Chile: Springer International Publishing. 2014. p. 89-96.
- COSTA, J. M. R.; FEITOSA, R. M.; DE SOUZA, C. R. B. RaisAware: Uma Ferramenta de Auxílio à Engenharia de Software Colaborativa Baseada em Análises de Dependências. **Scientia (Unisinos)**, v. 20, p. 12-24, 2009.
- COSTA, J. M. R.; SANTANA, F. W.; DE SOUZA, C. R. B. **Understanding Open Source Developers Evolution using TransFlow**. Proceedings of the 15th Collaboration Researchers' International Workshop on Groupware (CRIWG 2009). Peso da Régua, Portugal: Springer. 2009. p. 65-78.
- CROSS, R.; BORGATTI, S. P.; PARKER, A. Making Invisible Work Visible: Using Social Network Analysis to Support Strategic Collaboration. **California Management Review**, p. 25-46, 2002.
- DE SOUZA, C. R. B.; FROELICH, J.; DOURISH, P. **Seeking the source**: software source code as a social and technical artifact. Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work (GROUP '05). [S.l.]: ACM. 2005. p. 197-206.
- DE SOUZA, C. R. B.; MARCZAK, S.; PRIKLADNICKI, R. Desenvolvimento Colaborativo de Software. In: PIMENTEL, M.; FUKS, H. **Sistemas Colaborativos**. [S.l.]: Elsevier, 2011.
- DE VREEDE, G.; BRIGGS, R. **Collaboration Engineering**: Designing Repeatable Processes for High-Value Collaborative Tasks. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05). Washington: IEEE Computer Society. 2005. p. 1-10.
- DEFRANCO-TOMMARELLO, J.; DEEK, F. P. **Collaborative software development**: a discussion of problem solving models and groupware technologies. Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02). Hawaii: IEEE. 2002. p. 586-577.



- DEGRACE, P.; STAHL, L. H. **Wicked Problems, Righteous Solutions: A Catalog of Modern Engineering Paradigms.** [S.I.]: Prentice Hall, 1998.
- DEMARCO, T.; LISTER, T. **Peopleware: Productive Projects and Teams.** 2a. ed. [S.I.]: Dorset House Publishing Company, 1999.
- DIEHL, S. **Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software.** 1. ed. [S.I.]: Springer-Verlag New York, 2007.
- DOURISH, P.; BELLOTI, V. **Awareness and Coordination in Shared Workspaces.** Proceedings of the 1992 ACM conference on Computer-Supported Cooperative Work (CSCW'92). Toronto, Canada: ACM. 1992. p. 107-114.
- DYBA, T.; DINGSOYR, T. What Do We Know about Agile Software Development? **IEEE Software**, v. 26, n. 5, p. 6-9, Setembro 2009.
- FAN, S.; LI, X.; ZHAO, J. L. **Collaboration process patterns and efficiency of issue resolution in software development.** International Conference on Collaboration Technologies and Systems (CTS 2012). Denver: IEEE Press. 2012. p. 21-25.
- FISHER, D. A. **Social and Temporal Structures in Everyday Collaboration.** University of California. Irvine, USA. 2004.
- FISHER, D.; DOURISH, P. **ocial and Social and Temporal Structures emporal Structures in Everyday Collaboration in Everyday Collaboration.** UCI Irvine. [S.I.]. 2003.
- FREEMAN, L. C. Centrality in Social Networks: Conceptual clarification. **Social Networks**, v. 1, n. 3, p. 215-239, 1979.
- FSF. **The Free Software Definition. Disponível em: <http://www.gnu.org/philosophy/free-sw.html>. Acesso em: 12 Abril 2015.** [S.I.]. 2008.
- FUGGETTA, A. **Software process: a roadmap.** Proceedings of the Conference on The Future of Software Engineering (ICSE '00). New York: ACM. 2000. p. 25-34.
- FUKS, H. et al. **O Modelo de Colaboração 3C e a Engenharia de Groupware.** PUC-RIO. Rio de Janeiro. 2002. (ISSN 0103-9741).
- GAMMA, E. et al. **Design patterns: elements of reusable object-oriented software.** 1. ed. Boston, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

- GREENBERG, S. Computer-supported cooperative work and groupware: an introduction to the special issues. **International Journal of Man-Machine Studies**, 34, n. 2, Fevereiro 1991. 133-141.
- HAIN, S.; BACK, A. **State-of-the-Art on Maturity Models for Collaboration**. Universität St. Gallen. [S.l.]. 2009.
- HAINES, R.; VEHRING, N. **Increasing Team Coordination and Social Motivation Through Awareness Practices: A Case Study**. Proceedings of the European Conference on Information Systems (ECIS '12). Barcelona, Spain: [s.n.]. 2012. p. 1-12.
- HAKLAY, H.; SINGLETON, A.; PARKER, C. Web Mapping 2.0: The Neogeography of the GeoWeb. **Geography Compass**, v. 2, n. 6, p. 2011-2039, 2008.
- HERBSLEB, J. D. et al. **An empirical study of global software development: distance and speed**. Proceedings of the 23rd International Conference on Software Engineering (ICSE '01). Toronto, Canada: IEEE Computer Society. 2001. p. 81-90.
- HERBSLEB, J. D.; PAULISH, D. J.; BASS, M. **Global software development at Siemens: experience from nine projects**. Proceedings of the 27th International Conference on Software Engineering (ICSE 2005). New York: ACM. 2005. p. 524-533.
- HERBSLEB, J.; GRINTER, R. **Splitting the Organization and Integrating the Code: Conway's Law Revisited**. Proceedings of the 21st international conference on Software engineering (ICSE 99). New York: ACM. 1999. p. 85-95.
- HUMPHREY, W. S. **Managing the Software Process**. Boston, USA: Addison-Wesley, 1989.
- IBM. **IBM WebSphere Business Integration Monitor**. [S.l.]. 2003.
- IEEE. **International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering #2013; Software Life Cycle Processes #2013; Maintenance**. [S.l.], p. 1-46. 2006. (0-7381-4961-6).
- JANSEN, S.; BRINKKEMPER, S.; FINKELSTEIN, A. **Business network management as a survival strategy: A tale of two software ecosystems**. Proceedings of the 1st International Workshop on Software Ecosystems, 11th International Conference on Software Reuse. Falls Church, USA: [s.n.]. 2009. p. 34-48.

- JOHN-STEINER, V.; WEBER, R. J.; MINNIS, M. The Challenge of Studying Collaboration. **American Educational Research Journal**, v. 35, n. 4, p. 773-783, Dezembro 1998.
- JURISTO, N.; MORENO, A. M. **Basics of Software Engineering Experimentation**. [S.l.]: Springer Publishing Company, 2010.
- KADUSHIN, C. **Understanding Social Networks: Theories, concepts and findings**. 1ª. ed. [S.l.]: Oxford University Press, 2012. ISBN ISBN 978-0-19-537947-1.
- KANER, C.; BOND, W. **Software Engineering Metrics: What Do They Measure and How Do We Know ?** Proceedings of the 10th International Software Metrics Symposium (METRICS 2004). Chicago, USA: IEEE CS Press. 2004. p. 1-12.
- KANG, J. G.; HAN, K. H. **A Business Activity Monitoring System Supporting Real-Time Business Performance Management**. Proceedings of the Third International Conference on Convergence and Hybrid Information Technology (ICCIT '08). Busan, Coreia do Sul: IEEE. 2008. p. 473-478.
- KATZ, R.; ALLEN, T. J. Investigating the Not Invented Here (NIH) syndrome: A look at the performance, tenure, and communication patterns of 50 R & D Project Groups. **R&D Management**, v. 12, n. 1, p. 7-20, Janeiro 1982.
- KEDJI, K. A. et al. **Supporting collaborative development using process models: an integration-focused approach**. Proceedings of the International Conference on Software and System Process (ICSSP '12). Piscataway: IEEE Press. 2012. p. 120-129.
- KOCH, S. Software evolution in open source projects—a large-scale investigation. **Journal of Software Maintenance and Evolution: Research and Practice**, v. 19, n. 6, p. 361-382, 2007.
- KUSUMASARI, T. F. et al. **Collaboration model of software development**. International Conference on Electrical Engineering and Informatics (ICEEI 2011). Bandung: IEEE. 2011. p. 17-19.
- LEI, H. et al. **Contextual collaboration: platform and applications**. Proceedings of the 2004 IEEE International Conference on Services Computing (SCC 2004). Shanghai: IEEE. 2004. p. 197-206.
- LICCARDI, I.; DAVIS, H. C.; WHITE, S. **CAWS: An Awareness Based Wiki System to Improve Team Collaboration**. Proceedings of the Eighth IEEE International Conference on Advanced Learning Technologies (ICALT '08). Santander, Spain: IEEE. 2008. p. 265-267.

- LINDVALL, M. et al. **Empirical Findings in Agile Methods**. Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe 2002. Chicago, USA: Springer-Verlag. 2002. p. 197-207.
- LUNGU, M. et al. The Small Project Observatory: Visualizing Software Ecosystems. **Science of Computer Programming**, 75, n. 4, Abril 2010. 264–275.
- MACCORMACK, A.; RUSNAK, J.; BALDWIN, C. Y. Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. **Management Science**, v. 52, n. 7, p. 1015-1030, July 2006.
- MADEY, G.; FREEH, V.; TYNAN, R. **The open source software development**. Proceedings of the Americas Conference on Information Systems (AMCIS). Dallas: [s.n.]. 2002. p. 1806-1813.
- MADEY, G.; FREEH, V.; TYNAN, R. **The Open Source Software Development Phenomenon: An Analysis Based on Social Network Theory**. Proceedings of the Americas Conference on Information Systems (AMCIS2002). Dallas, USA: [s.n.]. 2002. p. 1806-1813.
- MAGDALENO, A. M. **Compootim: em direção ao planejamento, acompanhamento e otimização da colaboração na definição de processos de software**. COPPE/UFRJ. Rio de Janeiro. 2013.
- MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L. **A Roadmap to the Collaboration Maturity Model (CollabMM) Evolution**. Proceedings of the International Conference on Computer Supported Cooperative Work in Design (CSCWD). Lausanne, Switzerland: [s.n.]. 2011. p. 105-112.
- MAGDALENO, A. M.; ARAUJO, R. M.; WERNER, C. M. L. **An Exploratory Study on Collaboration Understanding in Software Development Social Networks**. Proceedings of the International Conference on Collaboration and Technology (CRIWG). Raesfeld: Springer. 2012. p. 113–120.
- MAGDALENO, A. M.; DE ARAUJO, R. M.; WERNER, C. M. L. **A roadmap to the Collaboration Maturity Model (CollabMM) evolution**. Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD). Lausanne: IEEE Press. 2011. p. 105-112.
- MALETIC, J. I.; MARCUS, A.; COLLARD, M. L. **A Task Oriented View of Software Visualization**. 1st International Workshop on Visualizing Software for

- Understanding and Analysis (VISSOFT). Paris, France: IEEE Computer Society. 2002. p. 32-40.
- MANGAN, M. A. S.; BORGES, M. R. S.; WERNER, C. M. L. Increasing Awareness in Distributed Software Development Workspaces. In: DE VREEDE, G.-J.; GUERRERO, L. A.; RAVENTÓS, G. M. **Groupware: Design, Implementation, and Use**. [S.l.]: Springer Berlin Heidelberg, v. 3198, 2004. p. 84-91.
- MASON, L. Fostering Understanding by Structural Alignment as a Route to Analogical Learning. **Instructional Science**, v. 32, n. 4, p. 293-318, 2004.
- MESSERSCHMITT, D. G. S. C. **Software Ecosystem: Understanding an Indispensable Technology and Industry**. Massachusetts, Cambridge: The MIT Press, 2003.
- MILI, R.; STEINER, R. Software Engineering - Introduction. In: DIEHL, S. **Revised Lectures on Software Visualization, International Seminar**. 1. ed. [S.l.]: Springer, 2002. Cap. 2, p. 129-137.
- MISTRÍK, I. et al. **Collaborative Software Engineering**. [S.l.]: Springer, 2010.
- MOCKUS, A.; FIELDING, R. T.; HERBSLEB, J. **A case study of open source software development: the Apache server**. Proceedings of the 2000 International Conference on Software Engineering (ICSE 2000). Limerick, Ireland: IEEE. 2000. p. 263-272.
- MOCKUS, A.; FIELDING, R. T.; HERBSLEB, J. D. Two case studies of open source software development: Apache and Mozilla. **ACM Transactions on Software Engineering and Methodology (TOSEM)**, v. 11, n. 3, p. 309-346, Julho 2002.
- MOHTASHAMI, M.; MARLOWE, T. J.; KU, C. S. Metrics are Needed for Collaborative Software Development. **Journal of Systemics, Cybernetics, and Informatics, Vol. 9 (5)**, , v. 9, n. Special Issue on Collaborative Enterprise, p. 41-47, Dezembro 2011.
- MOORE, C. R. Performance Measures for Knowledge Management. In: LIEBOWITZ, J. **Knowledge Management Handbook**. [S.l.]: CRC Press, 1999. Cap. 6.
- NETCRAFT. **February 2005 Web Server Survey Finds 40 Million Sites on Apache**. Netcraft. [S.l.]. 2005.
- NONAKA, I.; KONNO, N. The Concept of 'Ba': Building a Foundation for Knowledge Creation. **California Management Review**, v. 40, n. 3, p. 40-54, Spring 1998.

- NOSEK, J. T. The case for collaborative programming. **Communications of the ACM**, v. 41, n. 3, p. 105-108, 1998.
- OBJECT MANAGEMENT GROUP. **Software & Systems Process Engineering Meta-Model Specification**. [S.l.]. 2008.
- O'CONCHUIR, E. et al. **Exploring the Assumed Benefits of Global Software Development**. Proceedings of the International Conference on Global Software Engineering (ICGSE '06). Washington: IEEE Computer Society. 2006. p. 159-168.
- OLIVA, G. A. et al. **Characterizing Key Developers in a Development Group: A Case Study with Apache Ant**. Proceedings of the 18th CRIWG Conference on Collaboration and Technology. Lower Rhine, Germany: Springer Berlin Heidelberg. 2012. p. 97-112.
- OLIVEIRA, M. S. **Previa: Uma Abordagem para a Visualização da Evolução de Modelos de Software**. COPPE/UFRJ. Rio de Janeiro. 2011.
- O'REILLY, C.; BUSTARD, D.; MORROW, P. **The war room command console: shared visualizations for inclusive team coordination**. Proceedings of the 2005 ACM symposium on Software visualization (SoftVis '05). New York, USA: ACM. 2005. p. 57-65.
- ORTIZ-ARROYO, D. Discovering sets of key players in social networks. In: ABRAHAM, A.; HASSANIEN, A.-E.; SNÁSEL, V. **Computer Communications and Networks**. [S.l.]: Springer London, 2010. p. 27-47.
- PAASIVAARA, M.; LASSENIUS, C. Collaboration practices in global inter-organizational software development projects. **Software Process: Improvement and Practice**, v. 8, n. 4, p. 183-199, Março 2003.
- PANAS, T.; BERRIGAN, R.; GRUNDY, J. **A 3D metaphor for software production visualization**. Proceedings of the Seventh International Conference on Information Visualization. Seattle: IEEE Press. 2003. p. 16-18.
- PARNAS, D. L. On the criteria to be used in decomposing systems into modules. **Communications of the ACM**, 15, n. 12, Dezembro 1972. 1053-1058.
- PAULUS, P. B.; NIJSTAD, A. B. **Group Creativity: Innovation through Collaboration**. 1a. ed. [S.l.]: Oxford University Press, 2003.
- PINKERTON, B. **WebCrawler: Finding What People Want**. University of Washington. Washington, USA. 2000.

- POURSHAHID, A. et al. **Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools.** Proceedings of the 10th Workshop on Requirements Engineering (WER'07). Toronto: [s.n.]. 2007. p. 80-91.
- PRICE, B. A.; BAECKER, R. M.; SMALL, I. S. A Principled Taxonomy of Software Visualization. **Journal of Visual Languages and Computing**, 4, n. 2, 1993. 211-266.
- RAMASUBBU, N. et al. **Configuring global software teams: a multi-company analysis of project productivity, quality, and profits.** Proceedings of the 33rd International Conference on Software Engineering (ICSE '11). [S.I.]: ACM. 2011. p. 261-270.
- RAYMOND, E. S. **The Cathedral and the Bazaar.** 1a. ed. [S.I.]: O'Reilly Media, 2001.
- RECKER, J. C. Opportunities and constraints : the current struggle with BPMN. **Business Process Management Journal**, v. 1, p. 181-201, 2010.
- RILLING, J. et al. **Software Visualization - A Process Perspective.** Proceedings of the 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2007). Banff: [s.n.]. 2007. p. 10-17.
- ROMAN, G.-C.; COX, K. C. A Taxonomy of Program Visualization Systems. **IEEE Computer**, 26, n. 12, Dezembro 1993. 11-24.
- ROSENTHAL, S.; FINGER, S. **Design Collaboration in a Distributed Environment.** 36th Annual Frontiers in Education Conference (FIE 2006). San Diego: IEEE. 2006. p. 13-18.
- ROZUM, J. A. **Concepts on Measuring the Benefits of Software Process Improvement.** [S.I.]. 1993.
- RUBIN, W. et al. **Process Mining Framework for Software Processes.** Proceedings of the International Conference on Software Process (ICSP '07). Minneapolis, USA: Springer Berlin Heidelberg. 2007. p. 169-181.
- SACHDEV, D. **Use Rational Performance Tester to monitor Collaborative Lifecycle Management server resources.** IBM. [S.I.]. 2013.
- SANTANA, F. W. **EvoITrack-Process: Uma ferramenta para visualização e acompanhamento da colaboração em processos de software.** Anais do IV Workshop de Teses e Dissertações do CBSOFT (WTDSOFT 2014). Alagoas: [s.n.]. 2014. p. 39-44.

- SANTOS, T. A.; ARAUJO, R. M.; MAGDALENO, A. M. **Bringing out collaboration in software development social networks**. Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement (Profes '11). New York: ACM Press. 2011. p. 18-21.
- SCACCHI, W. Process Models in Software Engineering. In: MARCINIAK, J. J. **Encyclopedia of Software Engineering**. 1. ed. New York, USA: John Wiley and Sons, Inc, 2001. p. 993-1005.
- SCHONENBERG, H. et al. Process Flexibility: a Survey of Contemporary Approaches. **Advances in Enterprise Engineering I**, Montpellier, v. 10, p. 16-30, 2008.
- SCHWABER, C. **The Changing Face of Application Life-Cycle Management**. Forrester Research Inc. [S.I.]. 2006.
- SHULL, F.; CARVER, J.; TRAVASSOS, G. H. **An empirical methodology for introducing software processes**. Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering (ESEC/FSE-9). New York: ACM. 2001. p. 288-296.
- SIEBEL, N. T. et al. Latitudinal and longitudinal process diversity. **Journal of Software Maintenance**, v. 15, p. 9-25, 2003.
- SOMMERVILLE, I. **Software Engineering**. 7<sup>a</sup>. ed. [S.I.]: Addison Wesley, 2004.
- STASKO, J. Handbook of Data Structures and Applications. In: MEHTA, D. P.; SAHNI, S. **Handbook of Data Structures and Applications**. 1. ed. [S.I.]: Chapman and Hall/CRC, 2004.
- STEINMACHER, I. et al. **Preliminary empirical identification of barriers faced by newcomers to Open Source Software projects**. Proceedings of the Brazilian Symposium on Software Engineering (SBES 2014). Alagoas, Brasil: IEEE. 2014. p. 51-60.
- SUNTINGER, M. et al. **The Event Tunnel: Interactive Visualization of Complex Event Streams for Business Process Pattern Analysis**. Viena. 2007.
- TACCA, M. C. Commonalities between Perception and Cognition. **Frontiers in Psychology**, v. 2, n. 358, 23 Setembro 2011.
- THOMSON, A. M.; PERRY, J. L.; MILLER, T. K. Conceptualizing and Measuring Collaboration. **Journal of Public Administration Research and Theory**, v. 19, n. 1, p. 23-56, December 2007.



- TRAINER, E. et al. **Bridging the gap between technical and social dependencies with Ariadne**. Proceedings of the 2005 OOPSLA workshop on Eclipse technology eXchange (eclipse '05). San Diego, USA: ACM. 2005. p. 26-30.
- TUROFF, M.; HILTZ, S. R. Computer Support for Group versus Individual Decisions. **IEEE Transactions on Communications**, 30, n. 1, 1982.
- VAHIA, C. M.; MAGDALENO, A. M.; WERNER, C. M. L. **EvoITrack-SocialNetwork**: Uma ferramenta de apoio à visualização de redes sociais para equipes de desenvolvimento de software. Congresso Brasileiro de Software: Teoria e Prática (CBSOFT) - Sessão de Ferramentas. [S.l.]: [s.n.]. 2011. p. 7--13.
- VALETTO, G. et al. **Using Software Repositories to Investigate Socio-technical Congruence in Development Projects**. Fourth International Workshop on Mining Software Repositories (ICSE Workshops MSR '07). Minneapolis, USA: IEEE Computer Society. 2007. p. 25-28.
- VAN DER AALST, W. P. et al. Business process mining: An industrial application. **Information Systems**, v. 32, n. 5, p. 713-732, July 2007. ISSN 0306-4379.
- VAN DONGEN, B. F. et al. The ProM Framework: A New Era in Process Mining Tool Support. In: CIARDO, G.; DARONDEAU, P. **Applications and Theory of Petri Nets**. [S.l.]: Springer Berlin Heidelberg, v. 3536, 2005. p. 444--454.
- VICKNAIR, C. et al. **A comparison of a graph database and a relational database: a data provenance perspective**. Proceedings of the 48th Annual Southeast Regional Conference (ACM SE '10). Kennesaw, Georgia: ACM New York. 2010. p. 252-258.
- VOGELSANG, L.; CARSTENSEN, P. H. **New Challenges for the Collaboration in Web-Based Information Systems Development**. Proceedings of the 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '01). Washington: IEEE Computer Society. 2001. p. 386-391.
- VON KROGH, G. et al. Carrots and rainbows: motivation and social practice in open source software development. **MIS Q.**, v. 36, p. 649-676, 2012.
- VON WATZDORF, S.; MICHAHELLES, F. **Accuracy of positioning data on smartphones**. Proceedings of the 3rd International Workshop on Location and the Web (LocWeb '10). Tokyo, Japan: ACM New York. 2010. p. 2-5.
- WASSERMAN, S.; FAUST, K. **Social Network Analysis: Methods and Applications**. Primeira edição. ed. Cambridge: Cambridge University Press, 1994.

- WERNER, C. M. L. et al. **EvoTrack**: A Plug-in-Based Infrastructure for Visualizing Software Evolution. Anais do I Workshop Brasileiro de Visualização de Software/II Congresso Brasileiro de Software: Teoria e Prática (CBSOFT). São Paulo: [s.n.]. 2011. p. 1-8.
- WERNER, C. M. L.; ARAUJO, R. M. D.; SANTORO, F. M. **CDSOFT: Balanceando Colaboração e Disciplina em Processos de Desenvolvimento de Software**. Rio de Janeiro. 2011.
- WHITE, H. C.; BOORMAN, S. A.; BREIGER, R. L. Social Structure from Multiple Networks: Blockmodels of roles and positions. **American Journal of Sociology**, v. 81, n. 4, p. 730-780, Janeiro 1976.
- WHITEHEAD, J. **Collaboration in Software Engineering: A Roadmap**. Future of Software Engineering (FOSE 2007). Washington: IEEE Computer Society. 2007. p. 214-225.
- WILLIAMS, L. **The Collaborative Software Process**. [S.l.]. 2000.
- WOHLIN, C. et al. **Experimentation in Software Engineering: An Introduction**. 1. ed. Norwell, USA: Kluwer Academic Publishers, 2000.
- YE, Y.; KISHIDA, K. **Toward an understanding of the motivation open source**. proceedings of the 25th International Conference on Software Engineering (ICSE '03). Washington, USA: IEEE Computer Society. 2003. p. 419-429.

# APÊNDICE A – Dados das Instâncias de Processos Emuladas

Este apêndice apresenta os dados criados para a avaliação do ambiente EvoTrack-Process.

Tabela 0.1. Atores e distribuição de papéis criados para o estudo.

Id	Ator	Papel	Sítio	
A1	April Sanchez	Suporte	Brasília	
A2	Anthony Nichols	Cliente	Porto Alegre	
A3	Daniela Angelo	Comercial	Rio de Janeiro	
A4	Helen Kelly	Desenvolvedor	Rio de Janeiro	
A5	Isabel Bleasdale		Rio de Janeiro	
A6	Jan Fisher		Rio de Janeiro	
A7	Joseph Hovell		Rio de Janeiro	
A8	Marc Marseau		Porto Alegre	
A9	Mauro Zanetticci		Porto Alegre	
A10	Giovanna Almeida		Porto Alegre	
A11	Favio Riviera		Responsável por Produto	Rio de Janeiro
A12	Michael Morrison		Qualidade	Rio de Janeiro
A13	Misa Kumagai			Porto Alegre

Tabela 0.2. Identificadores das atividades do processo e papéis associados.

Id	Tarefa	Papel
T1	Abrir Requisição de Mudança – Suporte	Suporte
T2	Abrir Requisição de Mudança – Cliente	Cliente
T3	Abrir Requisição de Mudança – Comercial	Comercial
T4	Descrever Tarefa	Responsável por Produto
T5	Descrever Detalhes Técnicos	Desenvolvedor
T6	Descrever Testes	
T7	Implementar Solução	
T8	Analisar Reprovação	
T9	Justificar Reprovação	
T10	Executar Testes	
T11	Descrever Reprovação	Qualidade
T12	Disponibilizar Alterações p/ Produção	

Tabela 0.3. Dados de execução das instâncias.

	Instância 1	Instância 2	Instância 3	Instância 4	Instância 5	Instância 6
T1	x	x	x	A2	x	x
T2	A1	x	x	x	x	x

	<b>Instância 1</b>	<b>Instância 2</b>	<b>Instância 3</b>	<b>Instância 4</b>	<b>Instância 5</b>	<b>Instância 6</b>
<b>T3</b>	x	A3	A3	x	A3	A3
<b>T4</b>	A10	A11	A11	A10	A11	A11
	A11					
<b>T5</b>	A4	A4	A4	A10	A4	A4
	A10		A5			
<b>T6</b>	A5	A4	A4	A8	A4	A4
	A8	A5	A5	A10	A5	A5
<b>T7</b>	A6	A5	A6	A5	A5	A6
	A7	A6	A7	A8	A6	A7
<b>T8</b>	A4	x	x	x	A4	A4
	A8					
<b>T9</b>	x	x	x	x	A4	A4
						A5
<b>T10</b>	A12	A12	A12	A13	A12	A12
<b>T11</b>	A12	x	x	x	A12	A12
<b>T12</b>	A12	A12	A12	A13	A12	A12

# APÊNDICE B – Instrumentos Utilizados na Avaliação do Ambiente EvoTrack-Process

*Este apêndice apresenta os instrumentos que foram utilizados durante a avaliação do ambiente EvoTrack-Process.*

## B. 1. Formulário de Consentimento

O termo de consentimento livre esclarecido foi 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 do Ambiente EvoTrack-Process

#### Termo de Consentimento Livre Esclarecido

##### a. Objetivo do Estudo

Este estudo visa avaliar visualizações de processos disponibilizadas no ambiente EvoTrack-Process em relação ao entendimento e acompanhamento da colaboração em um processo de desenvolvimento de software.

##### b. Idade

Eu declaro ter mais de 18 anos de idade e concordo em participar de um estudo conduzido por Francisco Werther Silva de Santana Júnior da COPPE/UFRJ.

##### c. Procedimento

Neste estudo, os participantes deverão realizar algumas tarefas com base nas visualizações apresentadas. Todos os documentos utilizados neste estudo serão apresentados ao participante e deverão ser preenchidos pelo próprio.

##### d. 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.

**e. 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**

Francisco Werther Silva de Santana Júnior

Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

**PROFESSORES RESPONSÁVEIS**

Prof<sup>a</sup>. Cláudia Maria Lima Werner

Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

Prof<sup>a</sup>. Andréa Magalhães Magdaleno

Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

Nome (em letra de forma): \_\_\_\_\_

Assinatura: \_\_\_\_\_

Data: \_\_\_\_\_

## B. 2. Formulário de Caracterização do Participante

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.

<b>Avaliação da Compreensão e Acompanhamento da Colaboração</b>	
<b>Formulário de Caracterização do Participante</b>	
<b>Código do Participante:</b>	

Este formulário contém algumas perguntas sobre sua experiência acadêmica e profissional.

### I. Formação acadêmica

- Pós-Doutorado
- Doutorado concluído
- Doutorado em andamento
- Mestrado concluído
- Mestrado em andamento
- Graduação concluída
- Graduação em andamento

### II. Área de Formação

---

---

---

### III. 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:

<b>Área de Conhecimento</b>	<b>Grau de Experiência</b>				
Processos de Software	0	1	2	3	4
Colaboração	0	1	2	3	4
Visualização de Software	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**

Inclua o número de meses de experiência para cada uma das áreas de conhecimento.

<b>Área de Conhecimento</b>	<b>Tempo de Experiência (meses)</b>
Processos de Software	
Colaboração	
Visualização de Software	

### **IV. Experiência no domínio exemplo**

Esta seção será utilizada para compreender o 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:

<b>Domínio</b>	<b>Grau de Experiência</b>		
Processos de Software	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.

### **V. Experiência em ambientes similares**

Esta seção será utilizada para compreender quão familiar você está com outras ferramentas de acompanhamento de processos. Por favor, indique o grau de experiência com este tipo de ferramenta seguindo a escala abaixo:



Ferramenta	Grau de Experiência		
Ferramentas de acompanhamento de processos	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.

*Desde já, agradecemos a sua colaboração.*  
Francisco Werther Silva de Santana Júnior  
Andréa Magalhães Magdaleno  
Cláudia Maria Lima Werner

### B. 3. Formulário para Realização do Estudo

O formulário para realização do estudo tem por objetivo apresentar as tarefas que devem ser realizadas pelo participante, bem como coletar as suas respostas.

<b>Avaliação da Compreensão e Acompanhamento da Colaboração</b>	
<b>Formulário para Realização do Estudo</b>	
<b>Código do Participante:</b>	

#### CONTEXTUALIZAÇÃO

Suponha uma situação onde você é o gerente de um projeto de desenvolvimento de software cujo objetivo é criar um produto de software 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 o ambiente EvolTrack-Process :

Etapa 1 – Questões 1, 2 e 3	
Horário de Início:	
Horário de Término:	

A1) Quais atores participaram da execução das instâncias **3** e **4**? Escreva o nome de cada um deles.

A2) Com quais outros atores de sítios diferentes o ator “**Isabel Bleasdale**” colaborou na instância **1** de processo? Escreva o nome de cada um deles.

A3) Quais atores já participaram da execução da atividade “**Implementar Solução**” em conjunto com o ator “**Jan Fisher**”? Escreva o nome de cada um deles.

Etapa 2 – Questões 4 e 5	
Horário de Início:	
Horário de Término:	

A4) Qual ator colaborou com o maior número de outros atores na instância **5** do processo?

A5) Quais atividades diferentes o ator **“Giovanna Almeida”** desempenhou na instância **1** em relação à instância **4**?

Etapa 3 – Questão 6	
Horário de Início:	
Horário de Término:	

***Utilizando os dados de quaisquer outras instâncias que você julgar apropriado:***

A6) Considerando a instância **6**, qual ator você indicaria para substituir o ator **“Helen Kelly”** durante a execução da atividade **“Descrever Testes”**? Por que?

*Obrigado pela sua colaboração.*

Francisco Werther Silva de Santana Júnior

Andréa Magalhães Magdaleno

Cláudia Maria Lima Werner

## B. 4. Questionário de Avaliação do Estudo

O questionário de avaliação do estudo deve ser entregue na última etapa do estudo, e tem como objetivo coletar as percepções e considerações sobre a execução das atividades com a utilização do ambiente EvoTrack-Process.

Avaliação da Compreensão e Acompanhamento da Colaboração	
Questionário de Avaliação do Estudo	
<b>Código do Participante:</b>	

Prezado(a) participante,

Esta é a última etapa do estudo de utilização do ambiente EvoTrack-Process. O objetivo deste questionário é obter informações adicionais para a avaliação do ambiente, 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) 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:

**5) Qual a maior dificuldade encontrada na realização das tarefas?**

**6) Como o ambiente EvoTrack-Process contribuiu para a realização das tarefas?**

- Facilitou bastante
- Facilitou um pouco
- Não teve muito impacto (indiferente / não facilitou em nada)
- Dificultou um pouco
- Dificultou bastante

Comentários:

**7) Como você classifica a contribuição das visualizações oferecidas pela EvoTrack-Process para a realização das tarefas?**

- Facilitou bastante
- Facilitou um pouco
- Não teve muito impacto (indiferente / não facilitou em nada)
- Dificultou um pouco
- Dificultou bastante

Comentários:

**8) Quais as funcionalidades do ambiente EvoTrack-Process que foram mais úteis na realização das tarefas?**

**9) Quais as funcionalidades do ambiente EvoTrack-Process que não contribuíram para a realização das tarefas?**



10) De acordo com sua opinião, liste os aspectos positivos da utilização do ambiente EvoTrack-Process.

11) De acordo com sua opinião, liste os aspectos negativos da utilização do ambiente EvoTrack-Process.

12) Você possui alguma sugestão para melhoria do ambiente EvoTrack-Process? Em caso positivo, por favor, especifique-a(s).

( ) Sim ( ) Não

Comentários:

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.*

Francisco Werther Silva de Santana Júnior  
Andréa Magalhães Magdaleno  
Cláudia Maria Lima Werner

## B. 5. Respostas das Tarefas do Estudo

A1) Quais atores participaram da execução da instância 3 e 4? Escreva o nome de cada um deles.

*Resposta: marc.marseau, mauro.zettici, giovanna.almeida, misa.kumagai, anthony.nichols*

*michael.morrison, helen.kelly, isabel.bleasdale, joseph.hovell, daniela.angelo, favio.riviera, jan.fisher*

A2) Com quais outros atores de sítios diferentes o ator “**Helen Kelly**” colaborou na instância 1 de processo? Escreva o nome de cada um deles.

*Resposta: marc.marseau*

A3) Quais atores já participaram da execução da atividade “**Implementar Solução**” em conjunto com o ator “**Jan Fisher**”? Escreva o nome de cada um deles.

*Resposta: isabel.bleasdale, joseph.hovell, mauro.zettici*

A4) Qual ator colaborou com o maior número de outros atores na instância 5 do processo?

*Resposta: isabel.bleasdale*

A5) Quais atividades diferentes o ator “**Giovanna Almeida**” desempenhou na instância 1 em relação à instância 4?

*Resposta: Instância 1: Descrever Tarefa; Descrever Detalhes Técnicos.*

*Instância 4: Descrever Tarefa, Descrever Detalhes Técnicos, Decrever Testes.*

A6) Considerando a instância 6, qual ator você indicaria para substituir o ator “**Helen Kelly**” durante a execução da atividade “**Descrever Testes**”? Por que?

*Resposta: Qualquer um, desde que o participante considere quem já desempenhou a tarefa antes e qual suas respectivas localidades geográficas.*

# ANEXO A – Modelo de Processo Utilizado no Estudo de Avaliação

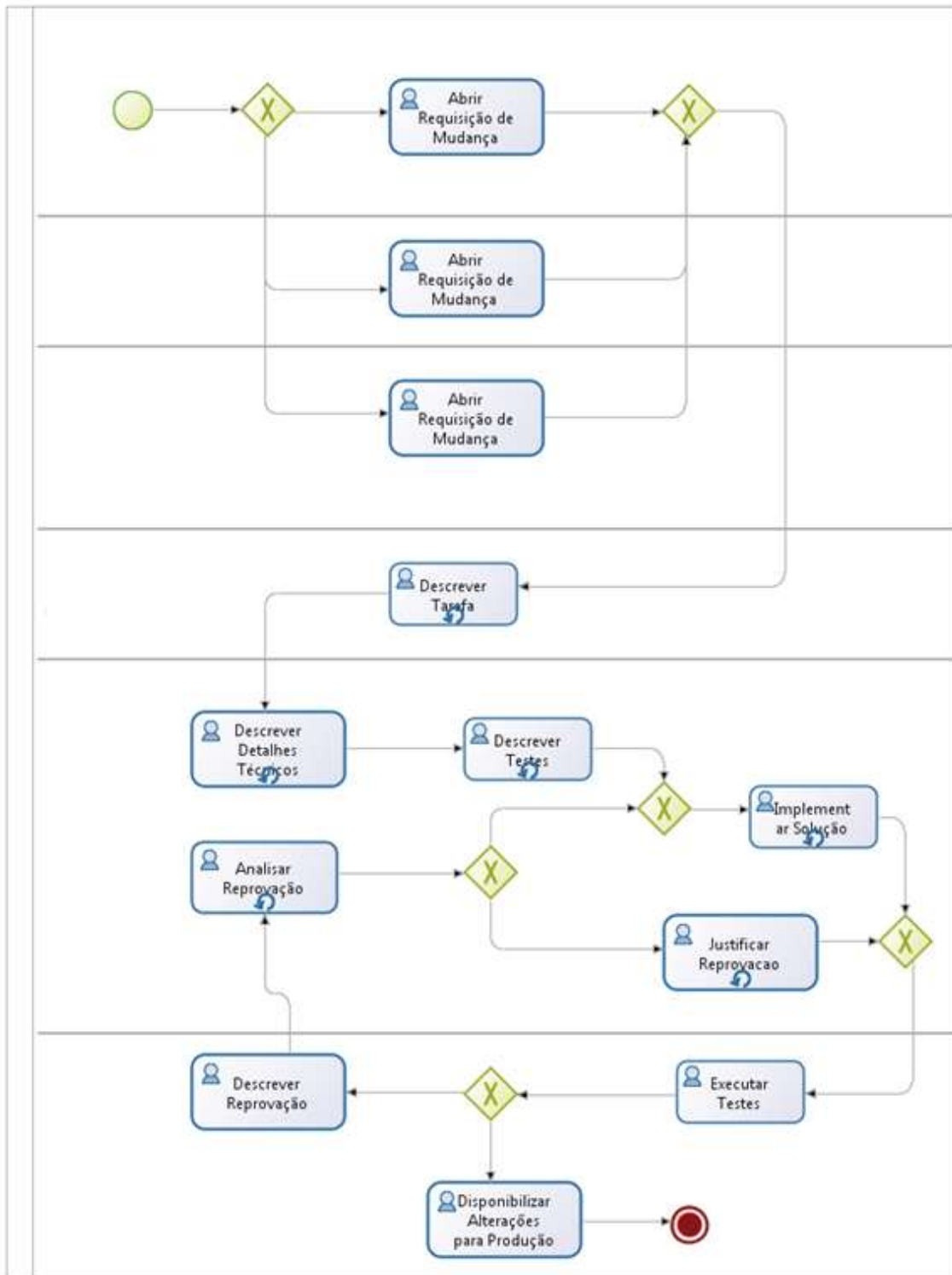


Figura A.1. Modelo de processos utilizado na avaliação.