



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



**UMA ESTRATÉGIA PARA O
DESENVOLVIMENTO DE HABILIDADES DE
COMUNICAÇÃO E COLABORAÇÃO PARA
FUTUROS ENGENHEIROS DE SOFTWARE**

Manaus - AM
20 de junho de 2013

LUIZ LEANDRO DOS REIS FORTALEZA

**UMA ESTRATÉGIA PARA O
DESENVOLVIMENTO DE HABILIDADES DE
COMUNICAÇÃO E COLABORAÇÃO PARA
FUTUROS ENGENHEIROS DE SOFTWARE**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Mestre em Informática.

ORIENTADORA: TAYANA UCHÔA CONTE
COORIENTADOR: RAFAEL PRIKLADNICKI

Manaus - AM
20 de junho de 2013

© 2013, Luiz Leandro dos Reis Fortaleza.
Todos os direitos reservados.

Fortaleza, Luiz Leandro dos Reis

Uma estratégia para o desenvolvimento de habilidades de
comunicação e colaboração para futuros engenheiros de software
/ Luiz Leandro dos Reis Fortaleza. — Manaus - AM, 2013
xxii, 87 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal do Amazonas
Orientadora: Tayana Uchôa Conte
Coorientador: Rafael Prikladnicki

1. Engenharia de Software. 2. Fatores Humanos.
3. Habilidades de Engenheiros de Software. 4. Desenvolvimento
Distribuído de Software. I. Título.

**UMA ESTRATÉGIA PARA O DESENVOLVIMENTO DE HABILIDADES
DE COMUNICAÇÃO E COLABORAÇÃO PARA FUTUROS ENGENHEIROS
DE SOFTWARE**

Luiz Leandro dos Reis Fortaleza

Dissertação de mestrado apresentada ao corpo docente do Instituto de Computação do Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Amazonas como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática.

Defesa em 20 de Junho de 2013, em Manaus, Amazonas, pela banca examinadora constituída pelos seguintes professores:

Prof^a. Tayana Uchôa Conte, D.Sc.

Prof. Adriano Bessa Albuquerque, D.Sc.

Prof. Marco Antônio Pinheiro de Cristo, D.Sc.

Prof. Rafael Prikladnicki, D.Sc.

Manaus – AM
20 de junho de 2013

Aos meus pais, Wanderley Rodrigues Fortaleza (in memoriam) e Ivaneide dos Reis Fortaleza.

Agradecimentos

A Deus, por ter me concedido a presente existência como meio de me levar a buscar minha própria evolução. Que eu faça bom uso desta e das oportunidades futuras.

À minha família pela compreensão, apoio, confiança, carinho, paciência e estímulo ao longo de minha vida acadêmica. Em especial à minha mãe, Ivaneide dos Reis Fortaleza, ao meu pai, Wanderley Rodrigues Fortaleza (*in memoriam*) e às minhas irmãs: Tatiana dos Reis Fortaleza Leite e Cristiana dos Reis Fortaleza.

À minha orientadora, Tayana Conte por ter confiado em mim para a realização deste trabalho e por ter me auxiliado nas principais decisões que levaram ao resultado final desta pesquisa.

Ao meu coorientador, Rafael Prikladnicki (PUCRS), por ter aceitado esta co-orientação e pelas diversas contribuições, fruto de sua *expertise* na área de Desenvolvimento Distribuído de Software (DDS).

À professora Sabrina Marczak (PUCRS) pela inestimável colaboração e co-autoria em um dos artigos gerados como fruto deste trabalho e ao professor Gleison Santos (UNIRIO) pelas ideias e contribuições fornecidas em uma das reuniões do Grupo de Pesquisa em Usabilidade e Engenharia de Software (USES).

Aos professores do PPGI-UFAM por terem contribuído para minha formação. Em especial ao Eduardo Nakamura que, por meio de suas aulas de Processo de Pesquisa em Ciência da Computação, me forneceu uma visão mais ampla sobre a pesquisa científica como um todo.

À *National Science Foundation* (NSF) e ao professor David Weiss (*Iowa State University*) por terem viabilizado minha participação no CTGDSD 2012, um *workshop* ICSE, o que me permitiu conhecer pessoalmente grandes pesquisadores da área de *Global Software Development*.

Aos membros do USES. Em especial ao Sérgio Roberto Costa Vieira e ao Olavo Olímpio Matos Júnior, pelo apoio na elaboração e condução dos estudos experimentais realizados.

Aos colegas de mestrado e doutorado do PPGI-UFAM, por terem me acompa-

nhado ao longo desta jornada. Em especial à Ludimila Gonçalves, Afonso Ribas, Adriana Penaranda, Alexandre Rodrigues, André Araújo, Diego Rodrigues, Viviane Gomes, Márcia Henke, Anna Beatriz Marques, Amadeu Anderlin Neto, Jacilane Rabêlo, Davi Viana dos Santos, Luís Jorge Rivero, Priscila Fernandes e Natasha Valentim.

Aos amigos Juan Gabriel Colonna e Jézika Cortez, pela amizade e companheirismo ao longo de todo o mestrado. Ambos foram fundamentais em momentos críticos, tanto os de cunho profissional quanto os de cunho pessoal.

Aos novos amigos que conheci nos dois últimos anos e que de alguma forma me forneceram excelentes *insights* durante conversas informais e que me estimularam à seguir em frente. Em especial ao Thawler Andrade, Valdemir Krause Junior (Jota), Cláudia Beeck, Aritz Carrera, Cintya Flores, Paolla Flessak (Lola) e Talita Cândido.

Aos amigos do Grupo Espírita Chico Xavier e da Fundação Allan Kardec. Em especial à minha turma de Estudo Sistematizado da Doutrina Espírita e aos amigos que trabalham comigo na atividade de vibração e irradiação, certamente os pensamentos positivos de todos vocês também me estimularam à seguir em frente.

Aos revisores anônimos que confiaram nos resultados obtidos desta pesquisa e que, com contribuições valiosas, possibilitaram a publicação dos mesmos em quatro conferências: WOSSES 2011, CTGDSD 2012, CSEE&T 2013 e SEKE 2013.

Aos alunos que voluntariamente participaram dos experimentos realizados.

Aos professores Adriano Bessa Albuquerque e Marco Antônio Pinheiro de Cristo por terem aceitado participar como membros da banca.

À CAPES pelo apoio financeiro concedido por meio de uma bolsa de mestrado.

Ao CNPq pelo apoio fornecido através do Projeto FTS-Brasil (Edital Universal, processo 483125/2010-5).

À coordenação do Programa Brasileiro de Qualidade e Produtividade de Software (PBQP-Software) por ter aprovado, durante o ciclo de 2012, o projeto **Incentivando o Desenvolvimento Distribuído de Software na Região Amazônica**, projeto que foi coordenado por mim e do qual esta pesquisa de mestrado faz parte.

Ao corpo administrativo do PPGI-UFAM pela presteza com que sanou eventuais problemas burocráticos ocorridos nos últimos anos.

“Nenhuma grande descoberta foi feita jamais sem um palpite ousado”
(Isaac Newton)

Resumo

A indústria de software precisa de profissionais que, além do conhecimento técnico, possuam determinadas habilidades que afetam positivamente o processo de desenvolvimento de software. Nesta dissertação de mestrado realizou-se inicialmente um mapeamento sistemático sobre quais habilidades são reportadas pela literatura como sendo mais importantes para o engenheiro de software. Este mapeamento serviu para limitar o escopo deste trabalho às habilidades de comunicação e colaboração.

O desenvolvimento de habilidades de comunicação e colaboração, segundo diversos autores, ocorre a partir de atividades práticas. Portanto, tentar capacitar melhor os alunos, no que diz respeito à comunicação e colaboração, seria integrá-los em projetos com a indústria. Todavia, nem sempre é possível estabelecer parcerias com a indústria de software. Uma alternativa é simular o cenário industrial em ambiente acadêmico.

Uma prática que tem crescido na indústria é a distribuição geográfica da equipe de desenvolvimento, conhecida como Desenvolvimento Distribuído de Software (DDS). Então, simular um cenário DDS é uma possível forma de aproximar os alunos da realidade da indústria de software. Esta é a abordagem proposta nesta dissertação.

Foram realizados dois estudos experimentais visando obter indícios sobre a eficiência da abordagem pedagógica proposta. A cada experimento buscou-se aprimorar a forma de coleta e análise dos dados, e efetuou-se a triangulação destes dados visando a obtenção de maior confiabilidade para as conclusões. Observou-se que a utilização de DDS é bastante motivadora para os alunos, todavia existem outros fatores que afetam esta motivação, como por exemplo a afinidade entre os membros das equipes co-locadas. Gerenciar adequadamente tais fatores, visando manter os participantes motivados, pode ser o diferencial entre sucesso e fracasso da atividade de ensino, uma vez que o desenvolvimento de habilidades é influenciado pela motivação.

Palavras-chave: Engenharia de Software, Fatores Humanos, Habilidades de Engenheiros de Software, Desenvolvimento Distribuído de Software.

Lista de Figuras

1.1	Elementos do desenvolvimento de software	1
1.2	Método de pesquisa	8
2.1	<i>Framework</i> Sócio-Cognitivo	11
2.2	Modelo MOCC simplificado	13
2.3	Temperamentos de Keirse-Bates	14
3.1	Artigos sobre habilidades ao longo dos anos	22
4.1	Exemplo de Sociograma representado por grafo	30
6.1	Representação gráfica da estratégia proposta	45
6.2	Exemplo de codificação aberta	47
6.3	Configuração das equipes no Estudo Experimental Preliminar	49
6.4	Resultados do estudo experimental preliminar	50
6.5	Configuração das equipes no Estudo de Observação	53
6.6	Exemplo de Interação no Estudo de Observação	53
6.7	Resultados do Estudo de Observação	54
6.8	Representação gráfica das respostas dos participantes	56
6.9	Representação do relacionamento entre os códigos identificados	57
6.10	Motivação e desmotivação durante o estudo experimental	57
6.11	Representação das relações entre códigos relativos às dificuldades percebidas	59

Lista de Tabelas

2.1	Fatores motivacionais da Teoria das Expectativas	12
2.2	Trabalhos sobre perfis psicológicos em Engenharia de Software	15
2.3	Níveis de Dispersão de Equipes de Desenvolvimento de Software	18
3.1	Objetivo principal do mapeamento sistemático	20
3.2	Termos de Busca	20
3.3	Mapeamento sistemático: síntese quantitativa	21
3.4	Mapeamento Sistemático: lista de habilidades e número de artigos associados	22
3.5	Competências de engenheiros de software “excepcionais” e “não-excepcionais”	23
4.1	Exemplo de matriz de competências	31
5.1	Alguns cursos de GSE mapeados	37
5.2	Estratégias de ensino de DDS global	38
5.3	Motivação em projetos acadêmicos com GSE	38
6.1	Objetivo do estudo de observação	52
6.2	Questões ligadas à satisfação e motivação	55

Sumário

Agradecimentos	ix
Resumo	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Motivação	2
1.2 Contextualização	3
1.3 Justificativa	4
1.4 Objetivos	6
1.4.1 Objetivo Geral	6
1.4.2 Objetivos Específicos	6
1.5 Método de Pesquisa	6
1.6 Estrutura do Documento	8
2 Fundamentação Teórica	9
2.1 O Fator Humano no Processo de Software e sua Influência no Desenvolvimento de Habilidades	9
2.1.1 Aprendizagem	10
2.1.2 Motivação	12
2.1.3 Personalidade	13
2.2 Ensino de Engenharia de Software	15
2.3 Desenvolvimento Distribuído de Software	16
2.4 Considerações Finais	18
3 Habilidades de Engenheiros de Software	19

3.1	Processo de Mapeamento de Habilidades	19
3.2	Discussão dos Resultados	21
3.3	Considerações Finais	24
4	Trabalhos Relacionados	25
4.1	Trabalhos sobre Colaboração	25
4.2	Trabalhos sobre Comunicação	27
4.3	Trabalhos sobre Ensino de ES	28
4.3.1	Utilização de DDS para o Ensino de ES	30
4.4	Trabalhos sobre Avaliação de Habilidades em Engenharia de Software .	31
4.5	Considerações Finais	32
5	Cursos de <i>Global Software Engineering</i>	35
5.1	Processo de Mapeamento de Cursos	36
5.2	Resultados do Mapeamento de Cursos	36
5.3	Considerações Finais	40
6	Utilizando DDS para o desenvolvimento de habilidades de comunicação e colaboração	43
6.1	Estratégia Proposta	44
6.2	Estudos Experimentais	45
6.2.1	Planejamento e Métodos de Coleta e Análise de Dados	46
6.2.2	Estudo Experimental Preliminar	47
6.2.3	Experimentação da Estratégia Proposta	51
6.3	Considerações Finais	58
7	Conclusão	61
7.1	Epílogo	61
7.2	Contribuições	62
7.2.1	Caracterização de Habilidades de Engenheiros de Software	62
7.2.2	Caracterização de Cursos de GSE	62
7.2.3	Estratégia para a Promoção do Desenvolvimento de Habilidades de Comunicação e Colaboração	63
7.2.4	Levantamento de Indícios sobre Aperfeiçoamento de Habilidades com o uso da Estratégia Proposta	63
7.3	Limitações	64
7.4	Perspectivas de Trabalhos Futuros	64

Referências Bibliográficas	67
A Lista de artigos selecionados no mapeamento sistemático de habilidades	75
B Termo de Consentimento Livre e Esclarecido	81
C Roteiro da entrevista	83
D Questionário utilizado no Estudo Experimental	85
E Lista de Acrônimos e Abreviaturas	87

Capítulo 1

Introdução

O desenvolvimento de software pode ser visto como uma tríade composta por pessoas, tecnologia e processos [Curtis et al., 1995; Carver et al., 2008]. A Figura 1.1 ilustra estes três elementos do desenvolvimento de software, destacando o elemento humano, pois são as pessoas que criam e utilizam processos e tecnologias. Buscar compreender e gerenciar os aspectos humanos pode ser um diferencial para o sucesso das atividades de uma organização [da Silva & França, 2012; Santos et al., 2011; Sharp et al., 2009; Beecham et al., 2008].

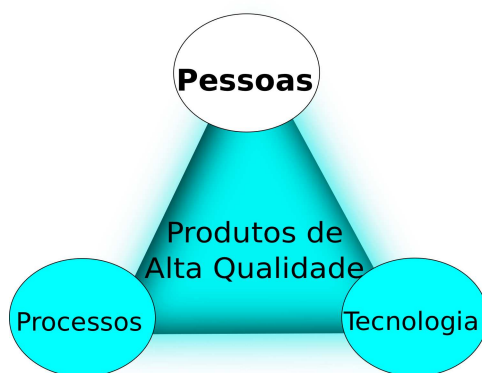


Figura 1.1. Elementos do desenvolvimento de software. Adaptado de Curtis et al. [1995].

Deste modo, para possuir este diferencial competitivo, a indústria necessita de profissionais tecnicamente qualificados e que possuam habilidades que lhes possibilitem desenvolver suas atividades de maneira eficiente. Isto é ilustrado por Akgün et al. [2007] que associaram a habilidade de trabalhar em equipe aos seguintes fatores: *speed to market*, menor custo de desenvolvimento e sucesso do produto no mercado.

Os cursos de Engenharia de Software, em geral, qualificam tecnicamente os alunos.

Entretanto promover o desenvolvimento de habilidades de comunicação, colaboração, e trabalho em equipe, em sala de aula, continua sendo necessário e ao mesmo tempo desafiador [Begel & Simon, 2008; Dawson, 2000].

Como não se identificou uma estratégia de ensino, amplamente aceita, que garanta o desenvolvimento de habilidades por parte dos alunos, identificou-se a oportunidade de se propor uma abordagem de ensino que busque estimular o desenvolvimento de tais habilidades por parte dos alunos envolvidos em cursos de Engenharia de Software. Com isto, busca-se a formação de engenheiros de software melhor capacitados para atender aos anseios da indústria de software.

Na abordagem pedagógica proposta nesta dissertação simulou-se a distribuição geográfica das equipes de desenvolvimento, pois alguns estudos apontam que utilizar a distribuição geográfica das equipes pode ser útil ao aperfeiçoamento de habilidades [Begel & Nagappan, 2008; Prikładnicki, 2011]. Deste modo, além de promover o desenvolvimento de habilidades de comunicação e colaboração, buscou-se levar os alunos a vivenciar uma modalidade de desenvolvimento que tem atraído bastante o interesse da indústria, o Desenvolvimento Distribuído de Software (DDS) [Carmel & Prikładnicki, 2010].

Nesta dissertação, apresentamos os resultados obtidos a partir de uma abordagem que procurou levar alunos de cursos de computação a desenvolver habilidades a partir de simulação de cenários reais de desenvolvimento. No restante deste Capítulo apresentamos a motivação (Seção 1.1), contextualização (Seção 1.2), justificativa (Seção 1.3), objetivos (Seção 1.4) e o método de pesquisa empregado (Seção 1.5) na condução desta pesquisa de mestrado.

1.1 Motivação

O fator humano pode ser decisivo para o sucesso ou fracasso de projetos de software [França & da Silva, 2010]. Dybå [2003] demonstrou que as pessoas têm um papel muito importante para o sucesso de iniciativas de melhoria do processo de software. Isto indica que ao se investigar questões relativas ao desenvolvimento de software faz-se necessário considerar o fator humano.

Conhecer as habilidades necessárias à execução das diferentes atividades envolvidas no ciclo de vida de software facilita a distribuição de papéis dentro de uma equipe [Martínez et al., 2010] e pode representar maior competitividade para uma empresa de desenvolvimento. Além disso, as características individuais dos membros da equipe podem afetar o desempenho da mesma.

Uma vez que as habilidades dos engenheiros de software influenciam no resultado final, faz-se necessário preparar os alunos dos cursos de ES para que os mesmos se tornem engenheiros de software mais bem preparados. **Mas de que modo é possível promover o desenvolvimento de habilidades não-técnicas por parte de alunos?**

Buscar formas de levar alunos a desenvolverem habilidades não-técnicas não é uma tarefa trivial, uma vez que habilidades como comunicação e colaboração são desenvolvidas, de modo geral, a partir da prática no próprio mercado de trabalho [Begel & Simon, 2008]. Contudo, ao se propor uma estratégia de ensino de Engenharia de Software que busque contemplar, além da técnica envolvida nas atividades de ES, o desenvolvimento de habilidades não-técnicas é possível contribuir para a formação de profissionais melhor preparados para a indústria de software.

Deste modo, as pesquisas relacionadas a habilidades de engenheiros de software são relevantes tanto em ambiente acadêmico quanto em ambiente industrial. O conhecimento sobre habilidades pode ser aplicado em ambiente industrial ou servir de base para a investigação de outros fatores humanos da Engenharia de Software por parte da academia. Além disso, ainda no ambiente acadêmico, existe a questão de como promover o desenvolvimento de habilidades por parte dos alunos.

1.2 Contextualização

Como mostrado na Seção 1.1, as habilidades dos profissionais de Engenharia de Software constituem um fator de importância dentro de ES e promover o desenvolvimento de habilidades pode ser um diferencial para a formação de um bom engenheiro de software. Entretanto, para definir quais habilidades devem ser ensinadas, é preciso descobrir quais são relevantes para as atividades de ES. Neste sentido a literatura reporta diversos trabalhos sobre habilidades [Akgün et al., 2007; Li et al., 2010; Turley & Bieman, 1995], todavia este conhecimento encontra-se disperso.

Para propor uma estratégia de ensino que procure formar profissionais com as habilidades adequadas às atividades de Engenharia de Software, faz-se necessário efetuar um levantamento de quais são estas habilidades. Tal levantamento pode ser feito através de um estudo secundário, que é um tipo de estudo caracterizado pelo uso de dados provenientes de trabalhos da literatura, com o objetivo de sintetizar as evidências [Sjøberg et al., 2007].

Nesta pesquisa realizou-se, como atividade inicial, um mapeamento sistemático [Kitchenham et al., 2011; Petersen et al., 2008], que é um tipo de estudo secundário,

objetivando caracterizar as habilidades de engenheiros de software. Esta atividade foi necessária para nortear a elaboração da estratégia de ensino proposta.

A partir do mapeamento de habilidades, elaborou-se uma estratégia de ensino objetivando responder a seguinte questão de pesquisa: **“Como promover o desenvolvimento de habilidades de comunicação e colaboração por parte de alunos de modo a melhor capacitá-los para atuar na indústria de software?”**. Para avaliar a viabilidade da abordagem proposta foi realizado um estudo piloto e um estudo experimental com duas turmas distintas.

No escopo desta pesquisa, considerou-se o engenheiro de software como sendo todo profissional envolvido diretamente com o desenvolvimento de software [Turley & Bieman, 1995], uma vez que não existe consenso a respeito dos papéis adotados pelo profissional de Engenharia de Software. Com relação à habilidade, tem-se que é um termo que vem do latim (*habilitas*) e refere-se à capacidade e disposição de executar algo. Segundo Asendorpf [2004] uma habilidade pode ser entendida como aquilo que uma pessoa executa com talento e destreza visando alcançar a um determinado objetivo.

Os indícios, obtidos a partir desta pesquisa, foram coletados com base na percepção dos alunos envolvidos, da mesma forma como ocorre em outras frentes de estudo do mesmo tema [Turley & Bieman, 1995; Begel & Nagappan, 2008; Rivera-Ibarra et al., 2010]. Estes alunos nunca haviam participado de projetos geograficamente distribuídos, deste modo também tiveram a oportunidade de utilizar uma modalidade de desenvolvimento que tem se tornado cada vez mais comum. Para obter uma compreensão mais profunda sobre os resultados obtidos, realizou-se uma análise qualitativa [Seaman, 1999] utilizando procedimentos de *Grounded Theory* [Strauss & Corbin, 1998].

1.3 Justificativa

Alguns autores [Dawson, 2000; Stiller & LeBlanc, 2002; Walker & Slotterbeck, 2002] sugerem que o engenheiro de software adquire as habilidades necessárias ao entrar em contato com um cenário de desenvolvimento real. Deste modo, inserir os alunos de ES em um ambiente real de desenvolvimento pode ser uma forma de estimular o desenvolvimento de habilidades.

Todavia, promover o contato dos alunos com empresas possui um custo elevado em razão da dificuldade de se conseguir parceiros industriais. Entretanto, para contornar esta dificuldade pode-se buscar a simulação de um ambiente real de desenvolvimento em sala de aula [Dawson, 2000].

O Desenvolvimento Distribuído de Software (DDS) é caracterizado por ter os membros da equipe de desenvolvimento localizados em locais geograficamente distantes durante o ciclo de vida do software, criando deste modo uma rede de sub-equipes [Jiménez et al., 2009]. Esta estratégia de desenvolvimento permite a uma empresa de desenvolvimento atender de maneira mais eficiente a diferentes mercados, além de possibilitar a divisão do esforço para o desenvolvimento de software. A utilização de DDS tem crescido na indústria [Carmel & Prikladnicki, 2010], de modo que utilizar DDS com alunos configura-se como uma forma de levar estes alunos a vivenciarem práticas utilizadas em situações reais de desenvolvimento de software.

Na abordagem de ensino proposta utilizou-se DDS porque esta modalidade de desenvolvimento possui muitos desafios relacionados a fatores humanos [Prikladnicki, 2009]. Além disso, segundo Richardson et al. [2007], quando se trabalha em ambiente distribuído, a linguagem, protocolos de comunicação, conhecimentos da cultura e de processos ganham maior importância do que em ambiente co-aloado, pois a necessidade de se desenvolver as mesmas se torna maior em função da distância. Estes fatores estão de certa forma relacionados às habilidades de comunicação.

Entretanto, a utilização de DDS, em geral, requer o estabelecimento de parcerias institucionais e o alinhamento pedagógico dos objetivos de ensino das instituições envolvidas o que pode despende um grande esforço de planejamento [Favela & Peña Mora, 2001]. Para contornar esta dificuldade optou-se por elaborar uma nova abordagem pedagógica na qual a distribuição das equipes de desenvolvimento é simulada, de modo a se obter os mesmos benefícios da distribuição real, porém despendendo um menor custo de esforço.

Outro fator para a utilização de DDS em cursos de ES é a existência de indícios que apontam um crescente interesse das instituições de ensino por inserir a distribuição geográfica das equipes em seus cursos regulares de ES [Fortaleza et al., 2012], objetivando a formação de profissionais melhor capacitados para atuar em um mercado global.

Além disso, existem indícios de que a prática de DDS contribui para a formação de um engenheiro de software melhor capacitado [Begel & Nagappan, 2008; Prikladnicki, 2011]. Espera-se reforçar tais indícios a partir dos estudos experimentais realizados ao longo desta pesquisa de mestrado.

1.4 Objetivos

Nesta Seção, apresentamos o objetivo geral (Seção 1.4.1) e os objetivos específicos (Seção 1.4.2) desta pesquisa de mestrado.

1.4.1 Objetivo Geral

O objetivo geral desta pesquisa é propor e avaliar experimentalmente uma estratégia de ensino que busque formar engenheiros de software melhor preparados, em termos de habilidades de comunicação e colaboração, para a indústria de software atual.

1.4.2 Objetivos Específicos

Os objetivos específicos desta pesquisa estão enumerados a seguir:

1. Caracterizar habilidades de engenheiros de software;
2. Analisar abordagens de ensino utilizadas para o desenvolvimento de habilidades;
3. Caracterizar cursos de *Global Software Engineering* (GSE);
4. Elaborar uma estratégia de ensino para o desenvolvimento de habilidades de comunicação e colaboração;
5. Promover a utilização da abordagem pedagógica baseada em DDS, em sala de aula, com turmas que não tiveram experiência prévia com DDS;
6. Avaliar experimentalmente os benefícios trazidos pela estratégia proposta a partir da percepção dos alunos que utilizaram DDS como parte de suas atividades acadêmicas.

1.5 Método de Pesquisa

O início do processo de pesquisa se deu com um mapeamento sistemático, que pode ser visto como um estudo secundário. Um estudo secundário é caracterizado pelo uso de dados provenientes de trabalhos da literatura com o objetivo de sintetizar as evidências [Sjøberg et al., 2007].

Um mapeamento sistemático da literatura [Petersen et al., 2008] pode ser visto como uma revisão sistemática de caráter exploratório [Kitchenham et al., 2011]. Em mapeamentos sistemáticos procura-se reunir todos os materiais relativos à questão investigada, incluindo artigos de opinião.

Este tipo de revisão da literatura foi apresentado por Kitchenham & Charters [2007] e, fazendo uso de um protocolo (visando garantir a replicabilidade do processo), constitui um meio de identificar, avaliar e interpretar materiais relevantes a uma determinada questão de pesquisa.

Após a realização do primeiro mapeamento sistemático, que teve como objetivo caracterizar habilidades de engenheiros de software, realizou-se uma análise de abordagens de ensino que buscaram promover o desenvolvimento de habilidade de comunicação e colaboração, pois estas foram as habilidades consideradas mais importantes. A seguir realizou-se um segundo mapeamento, desta vez com o objetivo de caracterizar cursos em que se utilizou desenvolvimento distribuído de software em cenários de distribuição global, *Global Software Engineering* (GSE). Este segundo mapeamento foi realizado utilizando a técnica *backward snowballing* [Webster & Watson, 2002], que é uma técnica na qual a partir de um conjunto inicial de evidências da literatura buscaram-se novas evidências a partir das referências utilizadas no conjunto inicial.

Além dos dois mapeamentos, o método de pesquisa utilizado ao longo deste mestrado é composto pela condução de experimentos com alunos de cursos de computação e análise da percepção destes alunos sobre o aperfeiçoamento de habilidades a partir da utilização de DDS. As fases do método de pesquisa adotado são descritas a seguir:

- **Caracterização de habilidades:** reunião de evidências da literatura sobre quais habilidades têm sido relatadas como sendo importantes para o engenheiro de software. Atividade executada com a realização de um mapeamento sistemático;
- **Análise de abordagens de ensino:** análise das abordagens de ensino que buscaram estimular o desenvolvimento de habilidades de comunicação e colaboração;
- **Caracterização de cursos GSE:** levantamento de cursos de Desenvolvimento Distribuído de Software com distribuição global (Global Software Engineering - GSE) objetivando capturar as principais características dos cursos que já foram realizados;
- **Elaboração da estratégia de ensino:** com base no resultado dos dois mapeamentos sistemáticos, elaborar a estratégia de ensino a ser utilizada;
- **Estudos experimentais:** experimentação da estratégia proposta com uma turma de alunos, coletando dados sobre a percepção de aprimoramento de habilidades.

O método de pesquisa adotado ao longo desta dissertação de mestrado é ilustrado na Figura 1.2

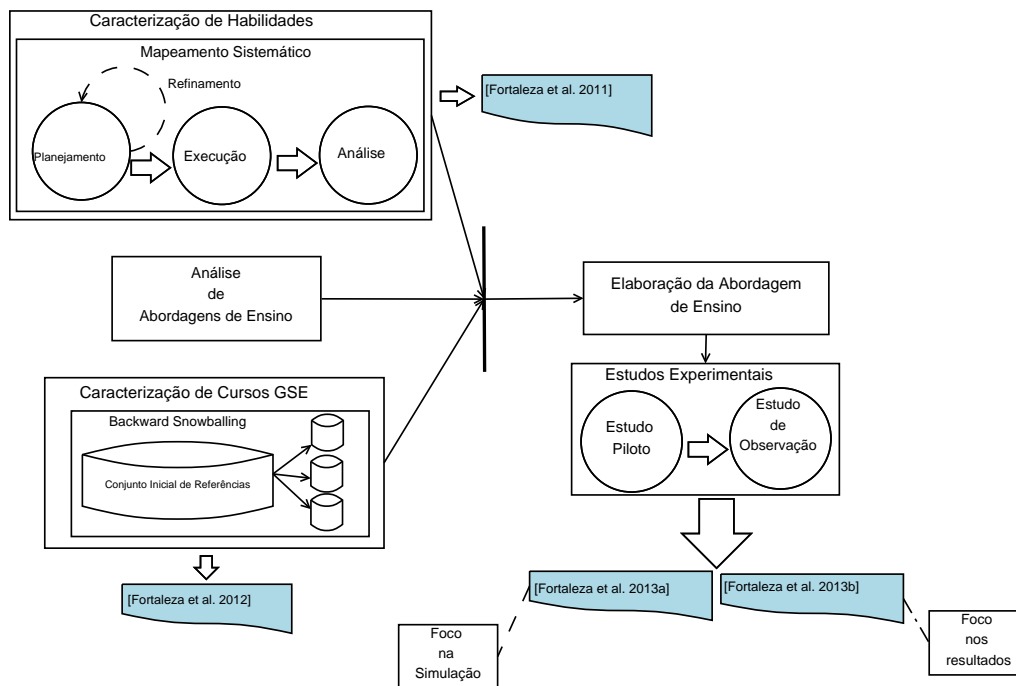


Figura 1.2. Método de pesquisa

Na fase de estudos experimentais, uma vez que existem indícios na literatura de que a utilização de DDS pode afetar positivamente o aperfeiçoamento das habilidades do indivíduo [Begel & Nagappan, 2008; Prikladnicki, 2011], ilustrando, portanto a viabilidade de se utilizar DDS com este fim, percebeu-se a oportunidade de se realizar estudos de observação seguindo a metodologia proposta por Shull et al. [2001]. Nesta metodologia, a fase de estudo de observação tem o propósito de verificar como um processo, reconhecidamente viável, ocorre.

1.6 Estrutura do Documento

A Introdução, contendo motivação, contextualização, objetivos e método de pesquisa, foi apresentada neste capítulo. No Capítulo 2 apresentamos a fundamentação teórica utilizada nesta proposta de pesquisa de mestrado. O Capítulo 3 apresenta o mapeamento sistemático de habilidades de engenheiros de software. O Capítulo 4 possui os principais trabalhos relacionados ao tema desta proposta. O Capítulo 5 apresenta o mapeamento de cursos de *Global Software Engineering*. O Capítulo 6 apresenta a estratégia proposta e os estudos experimentais realizados, com seus respectivos resultados. E, por fim, no Capítulo 7 são apresentadas as considerações finais, contribuições e perspectivas de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica utilizada nesta dissertação de mestrado. Com isto, são apresentados conceitos e terminologias utilizados ao longo do restante deste documento, de modo a possibilitar ao leitor uma maior compreensão do problema abordado.

Este capítulo encontra-se dividido em três seções principais que tratam de assuntos complementares. A Seção 2.1 trata dos fatores humanos envolvidos no processo de software; já a Seção 2.2 trata do Ensino de Engenharia de Software; e, por fim, na Seção 2.3 são apresentados conceitos relacionados ao Desenvolvimento Distribuído de Software (DDS), uma vez que a estratégia de ensino adotada faz uso desta modalidade de desenvolvimento de software.

Dentre os fatores humanos, buscou-se destacar àqueles que se relacionam mais diretamente com as habilidades de engenheiros de software, e estes são três: aprendizagem, motivação e personalidade.

2.1 O Fator Humano no Processo de Software e sua Influência no Desenvolvimento de Habilidades

Há mais de 30 anos discute-se a importância do fator humano para o desenvolvimento de software, entretanto em pesquisas mais recentes é possível notar que este fator ganhou importância ainda maior nos últimos anos [de Souza et al., 2009]. Portanto, pesquisas que exploram aspectos como comunicação e colaboração, tal como a apresentada nesta dissertação, são relevantes para a comunidade científica de Engenharia de Software.

Os fatores humanos podem explicar muitas falhas de sistemas de software

[Hazzan & Tomayko, 2004]. Portanto, compreender o aspecto humano da Engenharia de Software e saber como utilizá-lo pode ser um diferencial competitivo para as empresas de desenvolvimento de software. Além disso, várias pesquisas indicam que em iniciativas de melhoria do processo de software (MPS) o fator humano exerce um papel muito importante [Dybã, 2003; Matos Júnior et al., 2010; Santos et al., 2011].

Os diferentes aspectos do fator humano no processo de software são apresentados por Santos et al. [2011] com base na proposta de Robbins [2005]. Dentre estes aspectos, é possível destacar três que se relacionam com as habilidades do profissional de desenvolvimento de software, são eles: aprendizagem, motivação e personalidade. Os conceitos relacionados a esses três aspectos são apresentados a seguir:

- **Aprendizagem:** mudança no comportamento do indivíduo originada da experiência [McConnell, 1978];
- **Motivação:** um impulso ou sentimento que faz com que as pessoas ajam para atingir seus objetivos, podendo ser entendida como uma espécie de energia ou tensão psicológica que leva o indivíduo a agir [Penna, 2001];
- **Personalidade:** refere-se à forma como o indivíduo reage e interage com os demais, consistindo portanto em um conjunto de características que definem o modo como o indivíduo sente, pensa e se comporta [Atkinson et al., 1993].

Estes três aspectos relacionam-se com as habilidades do engenheiro de software, pois através da aprendizagem é possível adquirir uma nova habilidade e, a motivação e a personalidade influenciam neste processo, uma vez que sem motivação a aprendizagem pode não ocorrer [Mouly, 1973]. Logo, são fatores que afetam o processo de desenvolvimento de habilidades por parte do indivíduo.

2.1.1 Aprendizagem

“Aprendizagem é a progressiva mudança do comportamento que está ligada, de um lado, às sucessivas apresentações de uma situação e, de outro, aos repetidos esforços dos indivíduos para enfrentá-la de maneira eficiente”
[McConnell, 1978]

Para promover a aprendizagem podem ser utilizados diferentes métodos e técnicas de ensino. Uris [1966] agrupou os métodos de ensino em quatro categorias, que são listadas a seguir:

- Método de aprender pela experiência: neste método o indivíduo aprende com a experiência adquirida ao longo do tempo. Em outras palavras: “aprende fazendo”;
- Método conceitual: método de aprender pela teoria, ou seja, a partir de conhecimento de novos conceitos;
- Método simulado: método em que a aprendizagem ocorre através da imitação da realidade. Em outras palavras, é uma vivência simulada dos conceitos e teorias;
- Método de aprendizagem comportamental: método que se preocupa com a mudança comportamental do indivíduo com relação ao trabalho.

De acordo com Bíscaro [1994] não é possível afirmar que um grupo de métodos de ensino seja superior ao outro. Portanto, ao se identificar a necessidade de integrar os alunos à realidade da indústria de desenvolvimento, pode-se utilizar o método de aprender pela experiência, entretanto quando tem-se um cenário em que ocorre dificuldade de se estabelecer parcerias com a indústria, o método simulado pode ser uma boa alternativa.

A aprendizagem, de acordo com Downey [2005], é mais eficiente ao se utilizar situações práticas, pois de acordo com o *framework* proposto por este autor o indivíduo aprende mais a partir da observação de situações práticas. Este *framework* sócio-cognitivo é ilustrado pela Figura 2.1. Como é possível observar nesta figura, a vontade de aprender (motivação) relaciona-se diretamente com as habilidades e por isso na abordagem utilizada por esta pesquisa de mestrado decidiu-se coletar dados também a respeito da motivação dos alunos envolvidos.

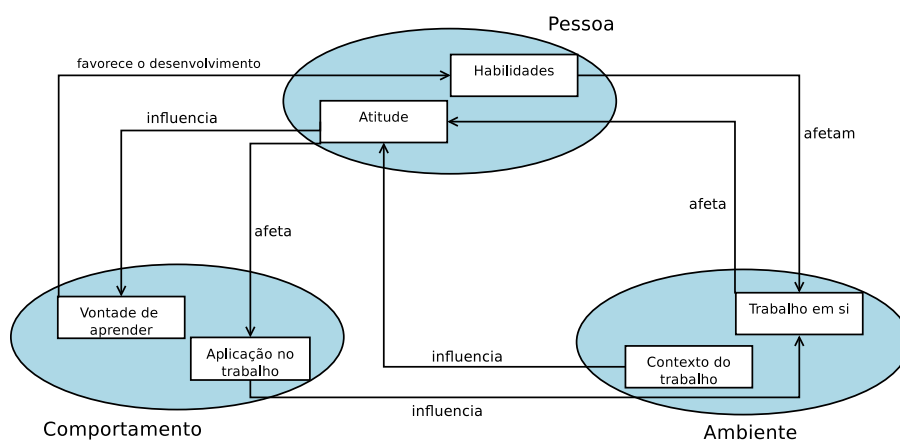


Figura 2.1. *Framework* Sócio-Cognitivo. Adaptado de Downey [2005].

O *framework* Sócio-Cognitivo foi elaborado com base em revisão de literatura e utilizando a Teoria Sócio-Cognitiva [Bandura, 1986]. No entanto, este *framework* não foi avaliado experimentalmente.

2.1.2 Motivação

No que diz respeito à motivação, França & da Silva [2010] citam duas teorias: a Teoria das Expectativas e a Teoria Motivação-Higiene. De acordo com estes pesquisadores a Teoria das Expectativas postula que a força motivacional é produto dos seguintes fatores: valor, expectativa e instrumentalidade, estes fatores são conceituados na Tabela 2.1. Já a Teoria Motivação-Higiene considera dois fatores: motivadores, que determinam o comportamento do indivíduo e podem aumentar a motivação; e, fatores higiênicos, os quais não causam motivação, mas quando não tratados adequadamente podem causar insatisfação.

Tabela 2.1. Fatores motivacionais da Teoria das Expectativas. Adaptado de França & da Silva [2010]

Fator	Conceito
Valor	Importância dada a uma recompensa.
Instrumentalidade	Estimativa feita sobre a obtenção de uma recompensa.
Expectativa	Relação entre esforço realizado e resultado obtido.

A motivação de engenheiros de software foi tema da revisão sistemática conduzida por Beecham et al. [2008], cujos resultados auxiliaram na criação do *Motivators, Outcomes, Characteristics and Context* (MOCC), que é um modelo de motivação apresentado por Sharp et al. [2009]. Este modelo utiliza a personalidade dos engenheiros de software como um elemento mediador.

Os elementos que compõem o MOCC são descritos da seguinte forma [Sharp et al., 2009]:

- Motivadores: fatores que motivam o engenheiro de software;
- Resultados: resultados da motivação dos engenheiros de software, tais como: produtividade, cumprimento de prazo, sucesso de projeto, dentre outros resultados positivos;
- Características: de acordo com o modelo MOCC, são aspectos identificados como característicos de engenheiros de software como por exemplo: capacidade técnica e criatividade;

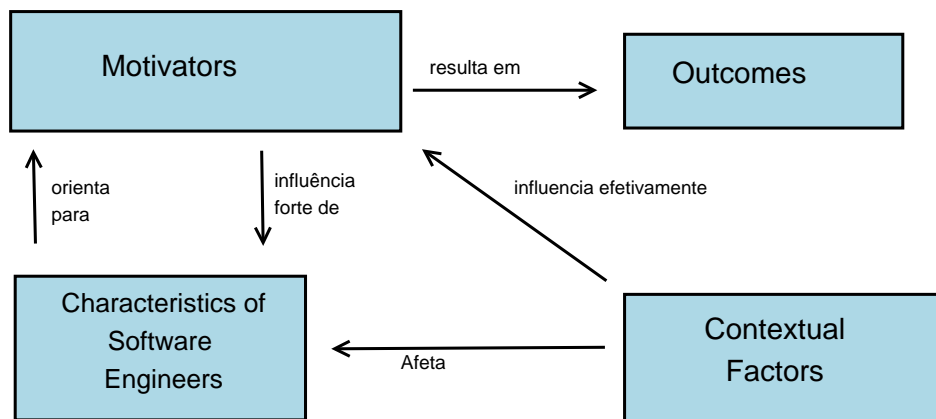


Figura 2.2. Modelo MOCC simplificado. Adaptado de Sharp et al. [2009].

- Contexto: fatores, como a personalidade do indivíduo e o ambiente, que afetam a motivação.

2.1.3 Personalidade

A personalidade relaciona-se com a motivação, com a aprendizagem e com as habilidades do engenheiro de software. Muitos trabalhos presentes na literatura preocupam-se em investigar o perfil psicológico enquanto fator de influência em Engenharia de Software [Teague, 1998; Ferreira & da Silva, 2008; Meira & da Silva, 2009; Capretz & Ahmed, 2010; Feldt et al., 2010; Martínez et al., 2010; Conte et al., 2011; Cruz et al., 2011].

A personalidade pode ser definida como sendo um conjunto de características que definem o modo como uma pessoa sente, pensa e se comporta [Feldt et al., 2010], ou seja, refere-se à forma como o indivíduo reage e interage com os demais. Da necessidade de se compreender os diferentes tipos de personalidade surgiram as teorias da personalidade e os indicadores psicológicos. As principais teorias utilizadas nas pesquisas sobre personalidade em Engenharia de Software estão listadas abaixo:

- *Myers-Briggs Type Indicator* (MBTI): instrumento para perfil psicológico individual que avalia a atitude das pessoas em relação ao mundo exterior, como gostam de receber informação, formam opinião e organizam suas vidas;
- *Five-Factor Model* [McCrae & Costa Jr, 2003]: modelo de tipos de personalidade baseado em cinco fatores: *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness* e *Neuroticism*;

- Temperamentos de *Keirsey-Bates* [Keirsey & Bates, 1984]: instrumento para medir perfil psicológico que está dividido em quatro dimensões (temperamentos) que são combinações das dimensões do MBTI;

Vale ressaltar que alguns dos modelos adotados utilizam alguns conceitos em comum. Por exemplo, o indicador MBTI faz uso de quatro dimensões: (i) como o indivíduo se energiza - Introversão (I) x Extroversão (E); (ii) como o indivíduo adquire informação - Intuitivos (N) x Sensoriais (S); (iii) como o indivíduo toma decisões - com base no sentimento (F) x com base na razão (T); (iv) como o indivíduo organiza sua vida - com um planejamento rigoroso (J) x sem muito planejamento (P). Deste modo a partir da combinação das quatro dimensões são obtidos os 16 tipos MBTI. Os temperamentos de *Keirsey-Bates* fazem uso de combinações entre os fatores MBTI. Isto é ilustrado pela Figura 2.3, na qual é possível observar a relação entre os 16 tipos psicológicos do MBTI e os quatro temperamentos.

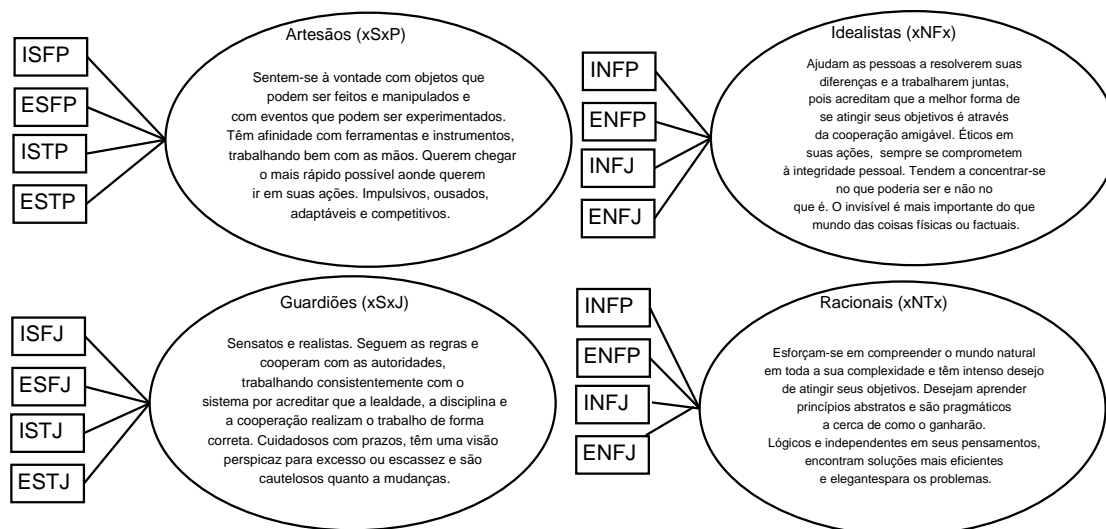


Figura 2.3. Temperamentos de Keirsey-Bates. Adaptado de Conte et al. [2011].

Os testes psicológicos categorizam a personalidade do indivíduo em alguma classe a partir de questionários desenvolvidos com este propósito. Isto possibilita conhecer o perfil psicológico de uma pessoa e, de posse desta informação, investigar a relação da personalidade com outros fatores, como por exemplo, habilidades. Uma lista dos principais trabalhos presentes na literatura que fizeram uso de testes de personalidade é mostrada na Tabela 2.2.

Tabela 2.2. Trabalhos sobre perfis psicológicos em Engenharia de Software

Referência	Descrição
[Paixão et al., 2012]	Estudo de possível relação entre perfil psicológico e evasão em cursos de Computação
[Conte et al., 2011]	Trabalho que investiga o perfil adequado a inspetores de usabilidade.
[Cruz et al., 2011]	Revisão sistemática sobre personalidade em Engenharia de Software.
[Capretz & Ahmed, 2010; Teague, 1998]	Estudos que associam atividades do processo de software a perfis psicológicos.
[Feldt et al., 2010]	Estudo que associa perfis psicológicos a atitudes de engenheiros de software.
[Martínez et al., 2010]	Estudo que investiga a influência da diversidade de personalidades no desempenho de equipes de desenvolvimento
[Meira & da Silva, 2009; Meira, 2009]	Estudos sobre perfis psicológicos adequados ao profissional de Garantia da Qualidade.
[Ferreira & da Silva, 2008]	Estudo sobre a influência dos perfis psicológicos na utilização de processo de software.

2.2 Ensino de Engenharia de Software

A Engenharia de Software apresenta diversos termos e conceitos, de modo que é possível elaborar um curso teórico que explore tais conceitos. Entretanto, tais abordagens teóricas podem nem sempre ser eficientes, pois com elas, em geral, não se têm indícios de que os alunos formados serão capazes de adotar os conceitos aprendidos em situações práticas.

Acredita-se que abordagens de ensino mais eficientes possivelmente são àquelas baseadas em projetos de desenvolvimento de software que, justamente por mostrarem aos alunos a aplicação do conhecimento teórico, os convencem de que tal conhecimento é importante [Stiller & LeBlanc, 2002]. Consequentemente, tais abordagens podem ser mais motivadoras para a aprendizagem.

Além disso, segundo Dawson [2000], o ambiente industrial apresenta certas situações que em geral não são vivenciadas nos cursos regulares de Engenharia de Software, como por exemplo, mudanças de prazo e requisitos descritos de forma ambígua. Este autor sugere que tais situações sejam inseridas propositalmente nos projetos acadêmicos de modo que os alunos possam vivenciar as pressões existentes em um cenário real de desenvolvimento visando maior preparo para o mercado de trabalho.

O processo de desenvolvimento de software é assimilado pelos alunos a partir de desenvolvimento sob condições similares às da indústria [Walker & Slotterbeck, 2002]. Ou seja, possivelmente a eficiência no processo de ensino/aprendizagem é obtida a partir da utilização de projetos práticos que reflitam as práticas da indústria de software.

Um engenheiro de software deve possuir tanto conhecimento técnico, habilidades

não-técnicas. Portanto, para a formação de bons profissionais, a academia deve buscar meios de promover o desenvolvimento de habilidades de comunicação e colaboração, além das habilidades técnicas que normalmente já são estimuladas nas abordagens tradicionais.

É possível que isto ocorra a partir de uma maior integração entre a indústria de software e a academia. Entretanto, quando se trata da formação do engenheiro de software ambos divergem: o mercado diz que a academia não tem feito um bom trabalho, enquanto que os pesquisadores da academia dizem que a indústria não utiliza as novas tecnologias geradas pela pesquisa acadêmica [Jazayeri, 2004]. Deste modo, estabelecer parceria com a indústria visando aperfeiçoar o ensino nos cursos de ES é uma solução interessante, porém não é trivial.

Além da utilização de projetos práticos, é importante estimular a colaboração entre os alunos, pois de acordo com Castro et al. [2008], independente da área, a colaboração leva as pessoas a desenvolver habilidades e elaborar estratégias para resolução de problemas e construção de conhecimento. Deste modo, estimular a colaboração entre os alunos em cursos de Engenharia de Software é uma possível forma de buscar promover o desenvolvimento de um conjunto maior de habilidades.

2.3 Desenvolvimento Distribuído de Software

Uma vez que a indústria de desenvolvimento de software tem distribuído geograficamente seus processos [Carmel & Prikladnicki, 2010], existem indícios de que cursos de Engenharia de Software tem buscado inserir projetos que fazem uso de Desenvolvimento Distribuído de Software (DDS) em seus programas de ensino [Fortaleza et al., 2012]. DDS permite que os membros da equipe de desenvolvimento estejam localizados em locais geograficamente distantes durante o ciclo de vida do software, criando deste modo uma rede de sub-equipes [Jiménez et al., 2009]. Deste modo, a adoção de desenvolvimento distribuído torna uma empresa mais competitiva, pois lhe permite atender diferentes mercados de forma mais eficaz e obter proveito de incentivos fiscais e mão-de-obra.

Além disso, Audy & Prikladnicki [2008] listam os seguintes fatores que levam as organizações a desenvolverem software de forma distribuída. De acordo com estes autores, estas razões são:

- Demanda e custo: a disponibilidade de recursos em diferentes localidades tornou-se um atrativo para a distribuição do processo de desenvolvimento;

- Rapidez de resposta ao mercado: a possibilidade de se obter vantagens de fuso-horário, pode possibilitar maior rapidez ao processo;
- Mercado e presença global: com centros de desenvolvimento espalhados em diferentes países;
- Rigor e experiência no desenvolvimento: equipes de desenvolvimento “co-localizadas” tendem a utilizar mais mecanismos informais. Já equipes distribuídas, tendem a melhorar significativamente a documentação e as ferramentas de comunicação, na tentativa de melhorar a comunicação;
- Sinergia cultural: a diversidade amplia a criatividade e a inspiração na organização de desenvolvimento de software;
- Escala: o crescimento de centros de desenvolvimento de software dificulta a tarefa de gerenciá-los, o que leva a distribuição do processo de desenvolvimento para atender à demanda.

Em razão das especificidades do cenário de desenvolvimento distribuído, Agerfalk et al. [2005] descrevem um conjunto de oportunidades e ameaças para esta modalidade de desenvolvimento. Os pesquisadores afirmam que tais desafios de DDS provêm de três dimensões: distância temporal, distância geográfica e distância sócio-cultural. Em virtude da distância, surgem dificuldades também nas atividades da Engenharia de Requisitos. Damian & Zowghi [2002] apontam as seguintes dificuldades relacionadas à distância geográfica dos *stakeholders*:

1. Comunicação inadequada;
2. Gerência ineficiente de conhecimento;
3. Diversidade cultural;
4. Diferença de fuso-horário

Com relação à distância geográfica entre os envolvidos, o entendimento do nível de dispersão em projetos DDS pode auxiliar a identificar fontes de dificuldades [Audy & Prikladnicki, 2008]. A Tabela 2.3 apresenta quatro possíveis níveis de dispersão em projetos de desenvolvimento de software que foram citados por Audy & Prikladnicki [2008].

Como evidenciado nesta seção, DDS oferece vantagens competitivas para as empresas que desejam utilizá-lo, todavia também fornece desafios, configurando-se em

Tabela 2.3. Níveis de Dispersão de Equipes de Desenvolvimento de Software

Nível	Características
Mesma localização física	Todos os atores em um mesmo local. Reuniões ocorrem sem dificuldade. Não existe diferença de fuso-horário
Distância nacional	Atores localizados no mesmo país. Pode haver diferenças de fuso-horário. Diferenças culturais em maior escala
Distância continental	Atores em países diferentes do mesmo continente. Reuniões face-a-face são dificultadas. O fuso-horário pode dificultar algumas interações
Distância global	Atores em países diferentes e em diferentes continentes. Comunicação e diferenças culturais podem ser barreiras para o trabalho. O fuso-horário exerce um papel fundamental

um cenário no qual as habilidades da equipe ganha um papel primordial. Faz-se necessário preparar os futuros engenheiros de software a atuarem neste tipo de cenário, no qual é possível que suas habilidades sejam aperfeiçoadas [Begel & Nagappan, 2008; Prikladnicki, 2011].

2.4 Considerações Finais

Neste capítulo foi apresentada a fundamentação teórica utilizada nesta pesquisa de mestrado. Uma vez que nessa pesquisa apresentamos uma estratégia de ensino de Engenharia de Software que visa estimular o desenvolvimento de habilidades por parte dos alunos, a fundamentação teórica inclui os aspectos humanos envolvidos em ES e o ensino de Engenharia de Software.

Como visto neste capítulo, o fator humano exerce grande influência no processo de software. As habilidades dos profissionais envolvidos é um fator cujo desenvolvimento é influenciado pela aprendizagem, motivação e personalidade.

Com relação ao ensino de ES, neste capítulo observou-se que às propostas mais eficientes são àquelas que buscam aproximar o ambiente acadêmico de um ambiente industrial, fornecendo aos alunos envolvidos nos cursos uma vivência mais próxima do desenvolvimento de software na indústria.

Também foram apresentados conceitos relativos ao DDS, pois a simulação da distribuição geográfica das equipes de desenvolvimento é a base da estratégia de ensino proposta nesta dissertação.

Capítulo 3

Habilidades de Engenheiros de Software

Este capítulo apresenta um levantamento sobre habilidades de engenheiros de software, que foi conduzido através de um mapeamento sistemático [Petersen et al., 2008], que é um tipo de revisão sistemática da literatura (RSL). Revisões sistemáticas da literatura têm o objetivo de reunir e sintetizar evidências a partir de um protocolo bem definido, o que visa garantir a replicabilidade do processo [Kitchenham et al., 2011]. Mapeamentos sistemáticos podem ser vistos como RSLs que possuem um escopo mais abrangente [Petersen et al., 2008].

No contexto desta pesquisa de mestrado, a execução deste mapeamento sistemático foi importante para auxiliar na limitação do escopo, pois a partir dos resultados obtidos é que foram escolhidas especificamente as habilidades de comunicação e colaboração a serem trabalhadas durante os estudos experimentais conduzidos.

No restante deste capítulo é apresentado o processo de mapeamento em si (Seção 3.1), a discussão dos resultados obtidos (Seção 3.2) e as considerações finais do capítulo (Seção 3.3)

3.1 Processo de Mapeamento de Habilidades

O mapeamento sistemático foi guiado por três questões de pesquisa. Estas questões são enumeradas a seguir, sendo a primeira questão a principal:

1. **Quais habilidades são relatadas na literatura como sendo importantes para o engenheiro de software?**
2. Qual o foco dos trabalhos que investigam habilidades de engenheiros de software?

3. Quais os tipos de artigos que tratam de habilidades de engenheiros de software?

O objetivo do mapeamento sistemático, descrito seguindo o paradigma GQM¹ [Basili & Rombach, 1988], é apresentado na Tabela 3.1.

Tabela 3.1. Objetivo principal do mapeamento sistemático

Analisar	trabalhos científicos encontrados por meio de uma abordagem sistemática
Com o propósito de	caracterizar
Com relação	habilidades necessárias ao profissional de desenvolvimento de software
Do ponto de vista	dos pesquisadores
No contexto	acadêmico (propostas curriculares de Engenharia de Software) e industrial

As evidências foram buscadas em quatro bibliotecas digitais: *IEEE Xplore*, *El Compendex*, *Scopus* e *ACM Digital Library*. Utilizamos três grupos de palavras que combinados formaram a *string* de busca. Na Tabela 3.2 são apresentados os termos associados aos respectivos grupos.

Tabela 3.2. Termos de Busca

Grupos	Termos
Desenvolvimento	Software Engineering, Software Development, Software Process, Software Project, Software Life Cycle.
Desenvolvedor	Software Engineer, Software Developer, Software Development Team, Software Engineering Team, Software Professional.
Habilidades	Skills, Abilities, Competencies, Qualification, Proficiency, Capacity, Aptness, Adeptness

A busca automática foi realizada sobre os meta-dados (título, abstract e *keywords*). A partir da combinação dos termos de busca obteve-se a seguinte string:

("software engineering"OR "software development"OR "software process"OR "software project"OR "software life cycle") AND ("software engineer"OR "software developer"OR "software development team"OR "Software Engineering team"OR "software professional") AND (skills OR abilities OR competencies OR qualification OR proficiency OR capacity OR aptness OR adeptness)

¹Goal Question Metric - abordagem sistemática utilizada em planos de medição

A estratégia de pesquisa utilizada é estruturada em três fases, que podem ser enumeradas da seguinte forma:

1. Leitura do título e do abstract de cada artigo retornado pela string de busca;
2. leitura da introdução e da conclusão dos artigos remanescentes do passo 1;
3. leitura integral dos artigos que passaram pelas duas primeiras fases de seleção.

A utilização do segundo filtro é uma possibilidade apontada por Kitchenham & Charters [2007] e que tem se mostrado útil em algumas revisões sistemáticas, como a realizada por Prikladnicki & Audy [2010]. Além disso, ao realizar o teste de protocolo constatou-se que somente a leitura dos abstracts seria insuficiente para filtrar adequadamente os artigos, para este contexto específico.

Após a leitura dos artigos aprovados foram incluídos também artigos citados como sendo base teórica, para isto efetuou-se um *backward snowballing* [Webster & Watson, 2002] das referências. Uma síntese quantitativa dos artigos selecionados é apresentada na Tabela 3.3, na qual são apresentados os resultados por fonte e fase de busca.

Tabela 3.3. Mapeamento sistemático: síntese quantitativa

Fonte	Início	Fase 1	Fase 2	Final
IEEE Xplore	52	26	19	11
El Compendex	53	23	15	9
Scopus	195	61	30	15
ACM DL	391	148	60	28
<i>Backward snowballing</i>	-	-	-	5
Total	691	258	124	68

Como é possível observar, ao final do processo obteve-se um total de 68 artigos selecionados. A lista completa dos artigos selecionados pelo mapeamento sistemático é apresentada no Apêndice A. Este mapeamento gerou uma publicação científica [Fortaleza et al., 2011].

O mapeamento sistemático também evidenciou a relevância do tema habilidades de engenheiros de software para a comunidade científica. Na Figura 3.1 é possível observar claramente um crescimento no número de publicações sobre este tema ao longo dos anos.

3.2 Discussão dos Resultados

As habilidades encontradas, através deste mapeamento sistemático e o respectivo número de artigos em que aparecem, são apresentadas na Tabela 3.4.

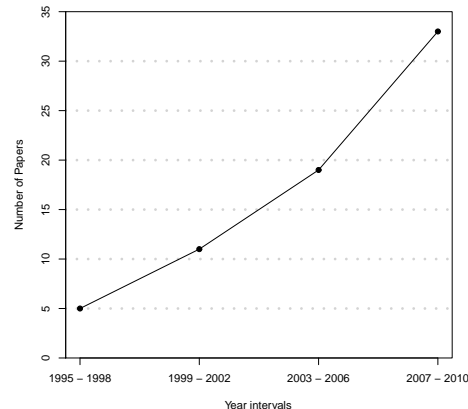


Figura 3.1. Artigos sobre habilidades ao longo dos anos

Tabela 3.4. Mapeamento Sistemático: lista de habilidades e número de artigos associados

Hab. de Comunicação	30	Pró-atividade	5	Organização	3
Hab. Técnicas	27	Resolução de conflitos	5	Comprometimento	3
Hab. de Trabalho em Equipe	22	Inovação	4	Lidar com pressão	3
Resolução de Problemas	12	Possuir visão ampla	4	Bom senso	2
Colaboração	10	Persistência	3	Persuasão	2
Liderança	10	Disciplina	3	Capacidade de Abstração	2
Flexibilidade	8	Sociabilidade	3	Gerenciar expectativas	1
Habilidades de Negociação	6	Criatividade	3	Atenção a detalhes	1
Pensar Criticamente	5	Auto-aprendizado	3	Raciocínio lógico matemático	1

Como é possível observar, as habilidades de comunicação figuram como as mais citadas, o que pode ser entendido como um indicativo da relevância deste tipo de habilidade para a Engenharia de Software. Em vista disso, optou-se por destacar alguns dos trabalhos que trataram de habilidades de comunicação. Uma vez que as habilidades de colaboração, em geral, são exploradas em conjunto com as de comunicação, e ambos os conjuntos de habilidades permeiam um conjunto maior de habilidades, decidiu-se incluir no escopo desta pesquisa as habilidades de colaboração.

Dentre os trabalhos selecionados no mapeamento sistemático destacamos o de Turley & Bieman [1995], pois apesar de ser um trabalho publicado há mais de 10 anos, continua a ser utilizado como referencial teórico em pesquisas atuais [Devlin & Phillips, 2010]. Neste estudo, os pesquisadores buscaram caracterizar habilidades de engenheiros de software, os quais foram agrupados em duas categorias: “excepcionais” - àqueles que apresentam produtividade acima da média; e, “não-excepcionais”, os que apresentam produtividade mediana. Este estudo foi conduzido em duas fases: a primeira com utilização do indicador MBTI e entrevistas; e, a segunda, através de um *survey*. As

competências associadas aos dois grupos de engenheiros de software são apresentadas na Tabela 3.5.

Tabela 3.5. Competências de engenheiros de software “excepcionais” e “não-excepcionais”. Adaptada de Turley & Bieman [1995].

Engenheiros de Software	Competências
Excepcionais	Ajudar os outros, pró-atividade com a gerência, exibir e articular convicções fortes, maestria de habilidades e técnicas, e possuir visão ampla
Não-excepcionais	Pedir ajuda, saber lidar com pressão, desejar contribuir e confrontar os outros

Como é possível observar algumas das competências descritas por Turley & Bieman [1995], como pedir ajuda e articular convicções fortes, estão associadas às habilidades de comunicação e colaboração. Além disso, em um trabalho no qual propuseram um *framework* para avaliação de habilidades, Rivera-Ibarra et al. [2010] caracterizaram um conjunto de competências que denominaram de competências sociais, às quais compreendem: relacionamento interpessoal, cooperação e trabalho em equipe, e, resolução de problemas. As habilidades de comunicação e colaboração também se relacionam a esse conjunto de competências.

De acordo com Largent & Lüer [2010], a comunicação e a colaboração são fatores decisivos para o sucesso do trabalho em equipe, e isto também é dito por Martínez et al. [2010] no trabalho em que investigaram o uso de medidas psicométricas para a definição de papéis em projetos de Engenharia de Software. O trabalho em equipe tem importância estratégica para as empresas de desenvolvimento, pois está associado ao sucesso do produto, velocidade de entrada no mercado, e redução no custo de desenvolvimento [Akgün et al., 2007].

Deste modo, observou-se que as habilidades de comunicação possuem uma relação com as habilidades de colaboração, e ambos os grupos permeiam o conjunto total de habilidades, incluindo habilidades técnicas, pois especificar corretamente um requisito, por exemplo, envolve habilidades e comunicação; e o trabalho colaborativo pode permitir a troca de conhecimento técnico [Pieterse et al., 2006]. Portanto, ao se estimular o aperfeiçoamento das habilidades de comunicação e colaboração, é possível que ao mesmo tempo se esteja aperfeiçoando um conjunto maior de habilidades.

Como o tema tratado nesta dissertação trata do desenvolvimento de habilidades em ambientes acadêmicos, julgou-se necessário destacar o estudo etnográfico conduzido por Begel & Simon [2008]. Neste estudo em questão, os pesquisadores investigaram

habilidades de engenheiros de software recém-formados, ou seja, que haviam saído recentemente do ambiente acadêmico.

Para a condução do estudo, Begel & Simon [2008] utilizaram entrevistas e observação *fly in the wall*² durante dois meses. Estes pesquisadores apontaram como pontos fortes dos observados: habilidades de codificação, leitura e escrita de especificação e criação de hipóteses. Além disso, registraram algumas ideias equivocadas dos participantes, tais como: dever fazer tudo sozinho, corrigir qualquer *bug* percebido por conta própria, pensar sempre que mais documentação faria diferença e pensar que consegue reconhecer quando se está estagnado na resolução de um problema.

3.3 Considerações Finais

Neste capítulo foram apresentados os procedimentos e resultados do mapeamento sistemático sobre habilidades de engenheiros de software. Foram coletados dados de 68 artigos, o que resultou em um conjunto de 27 habilidades.

Dentre estas habilidades, notou-se que as habilidades de comunicação e colaboração afetam outras habilidades, como a habilidade de trabalhar em equipe. Deste modo, optou-se por utilizar estes dois conjuntos de habilidades para delimitar o escopo da estratégia proposta nesta dissertação de mestrado.

Quando se tem um cenário com distribuição geográfica das equipes de desenvolvimento é possível que tanto comunicação quanto contribuição se tornem ainda mais desafiadoras. No capítulo seguinte são apresentados os principais trabalhos relacionados ao tema investigado nesta pesquisa de mestrado.

²Nesta técnica etnográfica, o observador apenas observa, buscando não interferir nas atividades rotineiras do observado

Capítulo 4

Trabalhos Relacionados

Este capítulo trata dos principais trabalhos sobre habilidades de comunicação e colaboração, ensino de Engenharia de Software, incluindo trabalhos que relatam experiências de uso de DDS em sala de aula, e também trabalhos sobre avaliação de habilidades. Procurou-se destacar a forma como tais pesquisas foram conduzidas e sintetizar seus principais resultados. Deste modo, buscou-se obter uma visão geral de quais caminhos já foram trilhados, pela comunidade científica de Engenharia de Software neste assunto.

O restante deste capítulo está organizado da seguinte forma: na Seção 4.1 são apresentados os trabalhos sobre habilidades de colaboração; na Seção 4.2, são apresentados trabalhos sobre habilidades de comunicação; na Seção 4.3 são apresentados trabalhos que versam sobre o ensino de Engenharia de Software; por fim, na Seção 4.4 são apresentados trabalhos que tratam da avaliação de habilidades.

4.1 Trabalhos sobre Colaboração

A colaboração desempenha um papel crucial para o desenvolvimento de software [Robillard & Robillard, 2000]. De acordo com Castro et al. [2008], a colaboração possibilita que as pessoas adquiram habilidades e desenvolvam estratégias para a resolução de problemas, contribuindo para a construção de conhecimento dentro de um determinado domínio. Deste modo, através da colaboração entre os membros de uma equipe de desenvolvimento o conhecimento acerca das especificidades de uma determinada aplicação é compartilhado e a confiança da equipe é reforçada.

De acordo com Whitehead [2007], a colaboração em Engenharia de Software atende a diferentes objetivos, dentre os quais é possível destacar os seguintes:

- Estabelecer o escopo do projeto;
- Elaboração da arquitetura e projeto;
- Gerenciamento da dependência entre as diferentes atividades;
- Registro das lições de um projeto a serem utilizadas em projetos futuros

Portanto, a colaboração está presente em todo o ciclo de vida do software. Isto pode ser visto como um indício da importância de buscar levar futuros engenheiros de software a desenvolver habilidades de colaboração ainda durante seus cursos de graduação.

Na pesquisa conduzida por Robillard & Robillard [2000], os pesquisadores conseguiram identificar quatro tipos de atividades relativas ao trabalho realizado em Engenharia de Software. Estes tipos de atividade estão listados a seguir, sendo as três primeiras atividades consideradas como atividades colaborativas:

- *Mandatory*: encontros formais, que seguem um planejamento prévio;
- *Called*: ocorre quando dois ou mais membros da equipe decidem unir-se para discutir uma determinada questão;
- *Ad hoc*: ocorre quando dois ou mais membros da equipe trabalham em um mesmo módulo e compartilham uma base *ad hoc* de conhecimento sobre a aplicação;
- *Individual*: ocorre quando um membro da equipe trabalha em uma tarefa específica de forma individual, sem compartilhar o conhecimento com os demais membros

Ao buscar a promoção do desenvolvimento de habilidades de colaboração, faz-se necessário planejar meios de estimular estes diferentes tipos de atividades. Deste modo, é possível pensar, por exemplo, na definição de marcos nos quais ocorram reuniões para a troca de informações e acompanhamento do projeto, o que caracteriza uma atividade do tipo *mandatory*. A inserção proposital de problemas pode ser um meio eficiente de levar os membros da equipe a interagir em atividades do tipo *called* e *ad hoc*, pois é possível que na busca por um maior entendimento os indivíduos tendam a interagir fora das reuniões previamente agendadas (marcos).

Em uma abordagem para o ensino de programação em grupo, Castro et al. [2008] sugerem que a partir de atividades individuais aos poucos sejam acrescentadas atividades colaborativas à medida que se aumenta a complexidade dos exercícios. Deste

modo, tem-se que o grau de complexidade de uma determinada atividade pode levar o indivíduo a buscar ajuda de seus pares.

Linhares et al. [2008] apresentaram um processo para nortear a negociação e colaboração em revisões formais de software. Este processo é dividido em quatro fases, que estão descritas a seguir:

1. Apresentação de propostas: os participantes apresentam possibilidades de solução;
2. Argumentação: os participantes defendem seus pontos de vista;
3. Negociação: os diferentes pontos de vista são confrontados. Nesta fase todos **colaboram** para que se atinja um objetivo comum;
4. Decisão: os participantes chegam a um consenso sobre o que deve ser feito.

Deste modo, as habilidades de colaboração estão intimamente relacionadas às habilidades de negociação. Portanto, é possível que ao estimular o desenvolvimento de habilidades de colaboração obtenha-se, indiretamente, ganho para habilidades de negociação. Ou seja, comunicação e colaboração permeiam um conjunto maior de habilidades.

Um exemplo de trabalho que busca caracterizar habilidades para um profissional específico é o que foi conduzido por Vale et al. [2011], no qual investigou-se habilidades relevantes para analistas de requisitos. Para isto, o autor realizou *surveys* tendo como participantes profissionais desta área de atuação e gerentes de projetos. Vale et al. [2011] também realizaram entrevistas semi-estruturadas, com utilização de procedimentos de *Grounded Theory* para análise dos dados coletados.

4.2 Trabalhos sobre Comunicação

Com base em uma pesquisa que teve a participação de 29 engenheiros de software de 22 companhias diferentes dos Estados Unidos, Ruff & Carter [2009], conseguiram identificar aspectos das habilidades de comunicação que são relevantes ao ambiente de trabalho do engenheiro de software. Dentre estes aspectos, estes pesquisadores destacaram os seguintes: planejar a comunicação de acordo com o contexto, clareza nas explicações, discutir produtivamente, saber ouvir, comunicar-se profissionalmente e dominar tecnologias e ferramentas mais utilizadas nos processos de comunicação. Isto ilustra que quando se promover o desenvolvimento de habilidades de comunicação em cursos de graduação, possivelmente se obtém profissionais melhor preparados às expectativas da

indústria de software, pois de acordo com os profissionais consultados nesta pesquisa a comunicação tem importância crucial para o processo de desenvolvimento de software.

Tomando por base o trabalho de Robillard & Robillard [2000], Cherry & Robillard [2008] conduziram um estudo etnográfico no qual identificaram diferentes papéis que o indivíduo pode assumir no processo de comunicação informal, ao longo de trabalhos colaborativos. No contexto da pesquisa conduzida por estes pesquisadores eles identificaram os seguintes papéis:

- *The rookie*: é um membro novo na equipe de desenvolvimento e que possui pouca experiência, portanto, inicia processos de comunicação visando esclarecer eventuais dúvidas;
- *The coach*: é o gerente de projetos, que ocupa a posição formal de líder. Não hesita em compartilhar informações que julgue relevante para o sucesso do projeto. É o responsável pela comunicação com os *stakeholders*;
- *The reference*: Mesmo não ocupando a posição oficial de liderança, é reconhecido pelos seus conhecimentos e experiência, o que faz com que seja consultado pelos demais membros da equipe para a resolução de problemas;
- *The cooperator*: é o indivíduo que independente, da posição que ocupe, está sempre pronto a colaborar com os demais colegas da equipe de desenvolvimento.

Após conduzir uma pesquisa com desenvolvedores de software ao longo de um ano, Pikkarainen et al. [2008] identificaram que a utilização de práticas ágeis - como *sprint planning* e *daily meeting* impacta positivamente na comunicação interna da equipe de desenvolvimento. Portanto, ao se utilizar práticas ágeis em projetos acadêmicos, tal como ocorreu no estudo piloto (Capítulo 6) que foi realizado nesta pesquisa de mestrado, é possível que se obtenha um impacto positivo sobre as habilidades de comunicação dos alunos envolvidos.

4.3 Trabalhos sobre Ensino de ES

Em um *survey* que teve a participação de 186 profissionais da indústria de software, Lethbridge [2000] buscou identificar quais conhecimentos são importantes para o engenheiro de software. Os resultados obtidos por este pesquisador apontam o inquestionável valor do aspecto técnico da Engenharia de Software, como, por exemplo, projetar algoritmos e possuir conhecimentos sobre tecnologias utilizadas na programação orientada a objetos, o que também é evidenciado por outros pesquisadores [Surakka, 2007],

mas também apontam que os profissionais consultados perceberam uma lacuna em sua formação no que diz respeito a aspectos humanos da Engenharia de Software, como a negociação.

Para cobrir esta lacuna na formação do profissional de software, surge a necessidade de se promover o desenvolvimento de habilidades ainda durante os cursos de graduação, o que pode representar uma redução nos custos de treinamento por parte da indústria de desenvolvimento.

Para preparar os alunos para o mercado, tornando-os profissionais melhor capacitados, Dawson [2000] sugere que certas situações, que ocorrem em ambientes reais de desenvolvimento, sejam inseridas propositalmente no ambiente acadêmico. Este autor elaborou, com base em sua experiência de ensino, uma lista contendo 20 exemplos de situações a serem trabalhadas em ambiente acadêmico e para cada exemplo uma sugestão de como trabalhar esta questão em sala de aula. Esta lista foi denominada pelo próprio autor como “20 truques sujos para treinar engenheiros de software”. Alguns exemplos de fatores a serem trabalhados com os alunos são: mudança de prazos, requisitos ambíguos, clientes com diferentes personalidades e mudança nos requisitos. Segundo Dawson [2000], estas são situações que ocorrem de forma recorrente no ambiente real de desenvolvimento e portanto se levarmos os alunos a vivenciá-las, estaremos contribuindo para uma maior qualidade de sua formação.

Callele & Makaroff [2006] sugerem que além de se trabalhar com um cenário próximo do desenvolvimento em ambiente industrial, deve-se ensinar Engenharia de Requisitos o mais cedo possível. Entretanto, ao tentar fazer isto, os pesquisadores concluíram que os alunos que participaram da experiência não possuíam maturidade suficiente para esse tipo de abordagem pedagógica, de modo que estes pesquisadores apontam como trabalho futuro a busca por uma estratégia pedagógica que contemple a Engenharia de Requisitos o mais breve possível, mas que tenha maior aceitação pelos alunos envolvidos. Dada a importância das atividades de Engenharia de Requisitos [Lemos & de Souza, 2008], optou-se por utilizar tais atividades nos estudos experimentais apresentados nesta pesquisa de mestrado.

Outros autores destacam a importância de se estimular o trabalho em equipe nas atividades em sala de aula, bem como considerar o perfil psicológico dos alunos envolvidos. Martínez et al. [2010] sugerem que se utilizem técnicas de psicometria e sociometria ao se formar equipes em trabalhos práticos, segundo estes pesquisadores isto aumenta as chances de formar uma equipe de sucesso. Dentre as técnicas de sociometria sugeridas por estes pesquisadores destaca-se a utilização de sociogramas, que podem ser entendidos como representações gráficas das relações sociais entre determinados indivíduos. Um exemplo de sociograma, representado através de um grafo, é

mostrado na Figura 4.1. Nesta figura as setas representam relações de afinidade entre os indivíduos identificados pelos círculos, podendo ser recíproca ou não, por exemplo: entre o indivíduo 2 e 6, tem-se afinidade recíproca, já entre os indivíduos 3 e 4, existe afinidade porém não de forma recíproca.

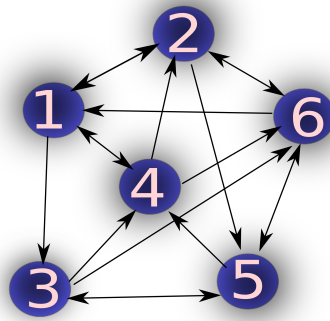


Figura 4.1. Exemplo de Sociograma representado por grafo

A partir de uma análise preliminar, que realizamos, da relação entre perfil psicológico e evasão em cursos de Computação, sugerimos que o perfil psicológico das turmas passe a ser uma variável a ser considerada na elaboração de novas estratégias de ensino e aprendizagem [Paixão et al., 2012]. É possível que com esta abordagem o número de evasões, em cursos de Computação, diminua.

Dada a importância do trabalho em equipe [Martínez et al., 2010; Largent & Lüer, 2010], nos estudos experimentais apresentados nesta dissertação de mestrado utilizou-se atividades práticas desenvolvidas por equipes de alunos. É possível que a partir da interação entre os membros das diferentes equipes, cada um possa por meio da troca de experiências, aumentar seus conhecimentos e aperfeiçoar suas habilidades, incluindo as habilidades de comunicação e colaboração.

4.3.1 Utilização de DDS para o Ensino de ES

De acordo com Begel & Nagappan [2008], a utilização de desenvolvimento geograficamente distribuído leva os profissionais de desenvolvimento a terem uma maior preocupação com os processos de comunicação e colaboração. De modo que tais habilidades podem ser mais bem desenvolvidas a partir da utilização de projetos DDS.

Almeida et al. [2012] relatam a experiência que tiveram a partir da utilização de DDS em um projeto envolvendo cinco universidades em cinco países diferentes. Segundo os autores, o curso tinha o objetivo de levar os alunos a colaborarem apesar das diferenças culturais e de fuso-horário, além de se configurar como um curso com uma

abordagem prática, na qual o aluno aprende através da experiência. Os pesquisadores afirmam que os alunos gostaram da experiência e recomendam que outras instituições passem a adotar DDS em seus cursos regulares.

A utilização de DDS em cursos de Engenharia de Software, segundo Prikladnicki [2011], leva os participantes a perceber a importância de competências e habilidades para a execução das atividades de desenvolvimento de software. Esta percepção os leva a tornarem-se engenheiros de software mais bem capacitados. Uma possível explicação, segundo o autor, está nos desafios inerentes à prática de DDS. O pesquisador coletou dados sobre a percepção do aperfeiçoamento de competências de 36 estudantes que atuaram em um projeto distribuído entre duas cidades do sul do Brasil. Os resultados obtidos são preliminares e ainda carecem de maior número de evidências experimentais, porém constituem indícios de que a utilização de DDS pode estimular o desenvolvimento de habilidades por parte dos alunos.

4.4 Trabalhos sobre Avaliação de Habilidades em Engenharia de Software

Uma investigação sobre formas de avaliar habilidades de engenheiros de software em ambiente acadêmico foi conduzida por Devlin & Phillips [2010] tendo por base as habilidades citadas por Turley & Bieman [1995]. Estes pesquisadores coletaram dados de estudantes distribuídos entre duas cidades do Reino Unido, e, para avaliação de habilidades fizeram uso de matrizes de competência, com auto-avaliação. Um exemplo de matriz de competência é apresentado na Tabela 4.1. Neste exemplo são exibidas competências relacionadas ao processo, estando cada uma dividida em três ranks, neste caso o indivíduo deve fornecer uma nota para cada um desses ranks.

Tabela 4.1. Exemplo de matriz de competências - Adaptado de Devlin & Phillips [2010]

Rank	1°	2°	3°
Processo	Passos necessários para a execução da tarefa		
Desempenho na execução da tarefa	Exibiu comportamento consistente durante a tarefa	Ajudou os outros a completar a tarefa	Incentivou os outros a executar a tarefa de forma independente
Habilidades de liderança	Aprendeu sobre estas habilidades	Ensaiou estas habilidades	Exerceu estas habilidades

Devlin & Phillips [2010] concluíram que o uso de matrizes de competências para avaliação de habilidades é promissor, todavia os alunos envolvidos no projeto talvez

não possuíssem maturidade suficiente para realizar auto-avaliação. Segundo os pesquisadores, a avaliação de habilidades é um campo de pesquisa a ser mais bem explorado.

Rivera-Ibarra et al. [2010] propuseram a adoção de um *framework* para a avaliação de competências. Para elaborar este *framework* foram coletados dados de estudos etnográficos realizados ao longo de sete anos. Estes pesquisadores agruparam as competências em três categorias: técnicas, sociais e pessoais. Eles sugerem um determinado nível de conhecimento para cada habilidade em função do papel desempenhado pelo engenheiro de software, definindo um percentual para cada nível, sendo este percentual definido em função da percepção do indivíduo. Estes níveis sugeridos são listados a seguir:

- 0 - menos de 25%;
- 1 - entre 25% e 50%;
- 2 - entre 50% e 75%;
- 3 - mais de 75%.

Apesar de apresentar um percentual para cada habilidade, no *framework* de Rivera-Ibarra et al. [2010] é o próprio indivíduo que atribui uma categoria a uma determinada habilidade, o que torna o processo subjetivo. Ou seja, da mesma forma que ocorre no trabalho de Devlin & Phillips [2010], a avaliação de habilidades é baseada na percepção do indivíduo, o que ocorre também em outros trabalhos similares [Guinan et al., 1998; Hogan & Thomas, 2005; Catanio, 2006].

Como é possível observar, a avaliação sobre o aperfeiçoamento de habilidades é realizada, de modo geral, com base na percepção do indivíduo. Deste modo, nos estudos experimentais realizados ao longo desta pesquisa de mestrado foram coletados dados sobre a percepção dos participantes no que diz respeito à aquisição ou aperfeiçoamento de suas habilidades de comunicação e colaboração.

4.5 Considerações Finais

Os trabalhos relacionados que foram apresentados ao longo deste capítulo tratam de habilidades de colaboração e comunicação de engenheiros de software, abordagens de ensino de Engenharia de Software, incluindo a utilização de Desenvolvimento Distribuído de Software em sala de aula, e também sobre avaliação de aquisição ou aperfeiçoamento de habilidades.

Percebeu-se que as habilidades de comunicação e colaboração afetam outras habilidades, como por exemplo habilidades de negociação e de trabalho em equipe. De modo que ao se trabalhar o desenvolvimento da comunicação e da colaboração, é possível que se esteja obtendo ganhos sobre um conjunto maior de habilidades. Entretanto, promover o desenvolvimento de habilidades por parte de alunos em cursos de Engenharia de Software ou similares continua sendo desafiador, pois, pelo que se observou durante o processo de pesquisa, diferente do que ocorre com as disciplinas tradicionais, não existe uma forma de promover habilidades que seja amplamente aceita.

Dado que o desenvolvimento de habilidades, em geral, ocorre a partir da experiência prática, pesquisadores sugerem que se procure colocar os alunos em contato com um cenário real de desenvolvimento, todavia muitas vezes isso não é possível. O que torna relevante as iniciativas de pesquisa que buscam novas formas de promover o desenvolvimento de habilidades.

Capítulo 5

Cursos de *Global Software Engineering*

Nesta pesquisa de mestrado busca-se promover o aperfeiçoamento de habilidades de engenheiros de software. O mapeamento sistemático apresentado no Capítulo 3 serviu para limitar o escopo para as habilidades de comunicação e colaboração. Nos capítulos anteriores também ficou evidenciado que muitos pesquisadores sugerem a adoção de projetos práticos para o desenvolvimento de habilidades. Uma alternativa é a utilização de projetos DDS.

Este capítulo discute as características, objetivos e desafios de cursos de *Global Software Engineering* (GSE). Realizou-se um mapeamento de tais cursos por entender-se que, quando a distribuição é global os desafios são maiores e, portanto tem-se, deste modo, um conjunto de lições aprendidas a serem aplicadas a cursos de Engenharia de Software que pretendam fazer uso de desenvolvimento geograficamente distribuído, mesmo nos casos em que a distribuição ocorra em menor escala.

Para realizar este mapeamento de cursos, optou-se por realizar um levantamento bibliográfico baseado em *backward snowballing* [Webster & Watson, 2002], que é uma técnica na qual a partir de um conjunto inicial de evidências da literatura buscam-se novas evidências a partir das referências utilizadas no conjunto inicial. Observou-se que existem evidências recentes [Jalali & Wohlin, 2012] que apontam a eficiência desta estratégia de levantamento bibliográfico, portanto identificou-se a oportunidade de experimentá-la no contexto desta pesquisa.

O conjunto inicial de trabalhos consultados foi gentilmente cedido por pesquisadores que desenvolveram uma pesquisa sobre a evolução e impacto da pesquisa em DDS no Brasil [Prikladnicki et al., 2011]. O restante deste capítulo apresenta o processo de mapeamento em si (Seção 5.1), os resultados do mapeamento de cursos (Seção 5.2) e

as considerações finais (Seção 5.3). Os resultados obtidos através do mapeamento de cursos também geraram uma publicação [Fortaleza et al., 2012].

5.1 Processo de Mapeamento de Cursos

O conjunto inicial de dados foi obtido a partir das seguintes bases de dados: *ACM Digital Library*, *Elsevier Science Direct*, *IEEE Xplore*, *SpringerLink* e *Wiley Interscience* [Prikladnicki et al., 2011]. Destas bases foram selecionados 12 artigos.

Deste conjunto inicial, seguindo a estratégia de busca *backward snowballing* [Webster & Watson, 2002] foram selecionados mais nove artigos, totalizando 21 artigos consultados. Entretanto, devido à limitação de páginas, muitos desses artigos deixaram de citar algumas informações sobre os cursos reportados, o que motivou a busca de tais informações nas páginas web das instituições e dos pesquisadores envolvidos. Durante esta busca em websites, também foram encontrados alguns cursos que não foram relatados em artigos científicos.

5.2 Resultados do Mapeamento de Cursos

Os cursos mapeados possuem características distintas, sendo alguns deles oferecidos de forma regular e contínua, enquanto outros são experiências realizadas de forma esporádica. Um exemplo do primeiro caso é a experiência realizada entre Croácia e Suécia, que resultou em um curso que é repetido anualmente desde 2003 [Crnković et al., 2003; Bosnić et al., 2011]. A lista completa de cursos mapeados é apresentada em [Fortaleza et al., 2012]. Na Tabela 5.1 são apresentados os países envolvidos nos cursos que utilizaram abordagens práticas e distribuição real das equipes.

Alguns cursos cobrem todas as fases de desenvolvimento de software enquanto outros se concentram em fases específicas, como elicitação e análise de requisitos [Audy et al., 2004; Damian et al., 2006].

A escolha por trabalhar com os alunos nas fases da Engenharia de Requisitos possivelmente é justificada por envolver atividades que são críticas para o desenvolvimento de software [Lemos & de Souza, 2008]. Além disso, trata-se de uma das primeiras fases do desenvolvimento de software, portanto quando se tem uma janela de tempo pequena prioriza-se as atividades desta fase.

Nas iniciativas de cursos de GSE além da parceria entre instituições de ensino é interessante também estabelecer parcerias com a indústria de desenvolvimento de software. Um exemplo disso é o *Global Studio Project* (GSP), que envolve diversas uni-

Tabela 5.1. Alguns cursos de GSE mapeados

Países envolvidos	Ano em que foi oferecido
Brasil e Estados Unidos	2002
Canadá, Austrália e Itália	2006
Estados Unidos e México	1997 e 1998
Suécia e Croácia	2003 a 2012
Alemanha e Rússia	Não informado
Estados Unidos, Índia e Cambodia	2005 a 2010
Estados Unidos e Alemanha	1997 e 1998
Brasil (duas cidades)	1997 e 1998
Alemanha (três cidades)	2010
Estados Unidos, Irlanda, Brasil, Alemanha e Índia	2004, 2005 e 2006
Estados Unidos e Turquia	2004
Suíça, Coreia, Vietnam, Ucrânia, Itália, Rússia, Hungria, Índia, Argentina e China	2007 a 2010
Estados Unidos e China	2011
Canadá e Finlândia	2012
Estados Unidos, China e Brasil	2012

versidades em parceria com a *Siemens* [Richardson et al., 2006; Mullick et al., 2006; Richardson et al., 2007]. A importância da interação com a indústria também é citada por Bellur [2006] que ao descrever sua experiência com ensino de GSE enfatiza a necessidade da utilização de projetos reais com clientes reais.

Com relação ao conteúdo abordado nos cursos, observou-se que, tal como os cursos tradicionais de Engenharia de Software, cobrem os seguintes tópicos:

- Elicitação, especificação e análise de requisitos;
- Projeto de sistemas;
- Ciclo de vida de software;
- Gerenciamento de projetos

Entretanto, cursos de GSE também se preocupam em fornecer conhecimento sobre a utilização de ferramentas colaborativas [Swigger et al., 2006], bem como em promover atividades que visam aperfeiçoar o trabalho em equipe, como atividades de socialização [Gotel et al., 2008]. Esta característica, trazida pela distribuição geográfica das equipes, possivelmente é o fator que torna a utilização de desenvolvimento distribuído uma forma de estimular o desenvolvimento e/ou aperfeiçoamento de habilidades de comunicação e colaboração.

Ainda com relação à comunicação e colaboração, Damian et al. [2006] afirmam que em projetos acadêmicos que utilizam desenvolvimento distribuído é necessário planejar adequadamente a interação, pois neste tipo de projeto a colaboração ganha uma importância ainda maior. Estes pesquisadores relataram a experiência em um curso conduzido através da colaboração entre instituições de três países: Canadá, Itália e Austrália, e com base nesta experiência elaboraram um conjunto de estratégias de ensino a serem utilizadas para aprimorar habilidades necessárias a GSE em ambiente acadêmico. Uma síntese destas estratégias é apresentada na tabela 5.2.

Tabela 5.2. Estratégias de ensino de GSE. Adaptado de Damian et al. [2006]

Habilidades	Estratégia de Ensino
Trabalho em equipe internacional	Times formados entre Universidades diferentes
Comunicação mediada por computador	Uso de diversas ferramentas de comunicação síncrona a assíncrona
Lidar com clientes remotos	Estudantes atuam nos dois papéis em dois diferentes projetos
Lidar com ambiguidade/incerteza	O grupo de alunos atuando como clientes define inicialmente os requisitos e ao longo do projeto são discutidas mudanças em função de incertezas e ambiguidades
Gerenciamento de projeto distribuído	Equipes auto-gerenciáveis

Outro fator observado a partir dos cursos mapeados diz respeito à motivação dos alunos envolvidos. A motivação pode significar o sucesso ou fracasso de uma iniciativa de ensino que utilize distribuição geográfica. Neste sentido, com base em uma experiência de mais de 10 anos utilizando GSE em projetos acadêmicos, Bosnić et al. [2011] listaram uma série de fatores motivantes e desestimulantes que devem ser trabalhados em cursos de GSE. Uma síntese desses fatores é apresentada na tabela 5.3.

Tabela 5.3. Motivação em projetos acadêmicos com GSE

Fatores Motivantes	Sucesso do projeto, notas, responsabilidade, suporte do cliente, conhecer novas pessoas/culturas, aprendizado de coisas novas, mudar de projeto e atmosfera da equipe
Fatores Desestimulantes	Atitudes pessoais, prazos rigorosos, excesso de documentação, questões de comunicação, sobrecarga de trabalho, falta de entusiasmo, questões técnicas, falta de tempo e baixa qualidade de trabalho

Para manter a motivação dos alunos Crnković et al. [2012] elaborou 10 dicas que visam o sucesso de um curso de GSE. Estas dicas estão enumeradas a seguir:

1. Criar situações que forcem a comunicação;
2. Promover a aproximação dos alunos envolvidos o mais cedo possível;

3. Manter o nível de comunicação sempre elevado, através de reuniões previamente planejadas, por exemplo;
4. Assegurar-se de que os estudantes tenham em mente a equipe geograficamente distante;
5. Manter os estudantes envolvidos;
6. Lembrar de que os grupos são diferentes, pois as características de cada grupo influenciam na motivação;
7. Ser flexível com relação às diferenças, pois uma vez que a diferença existe não se deve exigir características inerentes do outro grupo;
8. Ser flexível com relação à administração, pois diferentes intuições podem possuir diferentes regras;
9. Manter-se alerta;
10. Manter-se entusiasmado, pois a postura do professor pode influenciar os alunos.

Lições aprendidas da adoção de desenvolvimento distribuído em ambiente acadêmico também são reportadas por Purvis et al. [2004], que executaram um projeto envolvendo alunos da Nova Zelândia e da Alemanha. De acordo com estes autores, o software produzido utilizando GSE possui maior qualidade, e eles sugerem que isso se deve ao fato de que em projetos como esse existe maior motivação para a modularização dos componentes e necessidade de construção de *interfaces* mais claras, pois o sucesso de uma equipe depende do entendimento dos módulos fornecidos aos demais membros que estão dispersos. Os pesquisadores listam quatro recomendações para projetos de DDS em nível global, são elas:

1. Identificar o papel de cada membro da equipe e suas respectivas responsabilidades;
2. Adotar um conjunto de ferramentas de comunicação e desenvolvimento que seja aceito por todos os envolvidos;
3. Fornecer uma série de guias sobre os protocolos de comunicação ¹ e desenvolvimento;

¹Um protocolo de comunicação, neste caso, é o conjunto de regras que define como ocorrerá a comunicação

4. Ser flexível e adaptável quando novos requisitos surgirem.

Portanto, ao conduzir projetos com distribuição geográfica das equipes, em ambiente acadêmico, faz-se necessário estimular os fatores motivantes nos alunos envolvidos e buscar compreender os fatores desestimulantes visando mitigá-los por meio de ações efetivas, por exemplo: as questões de comunicação podem deixar de ser um problema ao se estabelecer claramente os protocolos de comunicação. Deste modo, quando se busca estimular o desenvolvimento de habilidades de comunicação e colaboração através do uso de DDS, faz-se necessário observar a motivação dos alunos envolvidos.

Pode-se concluir então que a utilização de distribuição geográfica das equipes pode despontar como uma forma eficiente de se promover o desenvolvimento de habilidades, entretanto nem sempre é possível estabelecer parcerias com indústrias de desenvolvimento ou mesmo instituições de ensino parceiras. Para contornar esta dificuldade uma possível solução é utilizar a simulação do cenário distribuído [Keenan et al., 2010].

5.3 Considerações Finais

Como demonstrado neste capítulo, os cursos de GSE têm aumentado ao longo dos anos [Fortaleza et al., 2012] e tais cursos tem como característica o fato de, além do conteúdo tradicionalmente abordado em Engenharia de Software, preocuparem-se com aspectos como comunicação e socialização dentre os membros das equipes que se encontram geograficamente dispersas.

Além disso, para que um projeto geograficamente distribuído tenha sucesso é preciso que seus participantes se mantenham motivados. Portanto, é importante coletar dados sobre a motivação dos envolvidos de modo a compreender melhor os fatores que levaram ao resultado final do projeto.

Dentro desta pesquisa de mestrado, a necessidade de mapear cursos GSE surgiu em razão do fato de o desenvolvimento de habilidades ocorrer a partir de projetos práticos, e cursos que utilizam DDS em geral apresentam uma abordagem prática. O mapeamento de cursos foi bastante útil para o planejamento dos experimentos conduzidos, pois a partir das lições aprendidas, das experiências relatadas na literatura, foi possível identificar possíveis problemas que poderiam surgir e mitigá-los. Além disso, pelo que foi apresentado dos cursos mapeados foi possível reforçar a importância das habilidades de comunicação e colaboração que foram enfatizadas no mapeamento sistemático de habilidades de engenheiros de software [Fortaleza et al., 2011].

Utilizar DDS em um ambiente acadêmico também tem a vantagem de fornecer aos alunos a vivência de uma experiência com uma modalidade de desenvolvimento que

tem atraído a indústria [Carmel & Prikladnicki, 2010]. O capítulo seguinte apresenta a estratégia de ensino proposta nesta dissertação e a a condução dos experimentos realizados com seus respectivos resultados.

Capítulo 6

Utilizando DDS para o desenvolvimento de habilidades de comunicação e colaboração

Neste capítulo é apresentada a estratégia de ensino proposta por esta pesquisa de mestrado, que tem por objetivo estimular o aperfeiçoamento de habilidades de comunicação e colaboração de alunos envolvidos em cursos de Engenharia de Software. Também são apresentados os experimentos realizados com a utilização desta estratégia.

Para delimitar quais habilidades seriam trabalhadas, foi realizado um mapeamento sistemático (Capítulo 3) no qual se observou que as habilidades de comunicação são as mais citadas e que este grupo de habilidades possui relação com as habilidades de colaboração. Tanto as habilidades de comunicação quanto as de colaboração permeiam outras habilidades, como por exemplo habilidades de trabalho em equipe e este foi o critério utilizado para delimitação deste escopo.

A literatura sugere que habilidades são desenvolvidas a partir da utilização de atividades práticas. Uma vez que os cursos de DDS, em geral, são desenvolvidos com a utilização de projetos práticos e existem estudos preliminares que apontam a utilização de DDS como forma de levar ao aperfeiçoamento de habilidades [Begel & Nagappan, 2008; Prikladnicki, 2011], optou-se por utilizar esta modalidade de desenvolvimento na estratégia proposta. Além disso, a prática de DDS tem atraído tanto à indústria [Carmel & Prikladnicki, 2010] quanto à academia [Fortaleza et al., 2012].

O restante deste capítulo encontra-se estruturado da seguinte forma: a Seção 6.1 apresenta a estratégia de ensino proposta; a Seção 6.2 apresenta os estudos experimentais conduzidos; e, por fim, a Seção 6.3 traz as considerações finais do capítulo.

6.1 Estratégia Proposta

A abordagem pedagógica a ser utilizada não deve ser apenas teórica, pois para o desenvolvimento de habilidades a utilização de atividades práticas mostra-se mais promissora [Dawson, 2000]. A partir dos resultados de um levantamento de cursos de *Global Software Engineering* [Fortaleza et al., 2012], constatou-se que cursos que fazem uso de Desenvolvimento Distribuído de Software de maneira geral fazem uso de projetos práticos. Também observou-se que o número de instituições interessadas em inserir DDS em seus cursos regulares tem aumentado ao longo dos anos e além disso as indústrias de desenvolvimento cada vez mais distribuem seus processos [Carmel & Prikładnicki, 2010], deste modo ao utilizar DDS é possível que se esteja fornecendo aos alunos a utilização de projetos práticos e a vivência de uma modalidade de desenvolvimento que está alinhada às expectativas da indústria de software atual.

Entretanto, implantar um curso de DDS não é uma tarefa trivial pois envolve o alinhamento de objetivos pedagógicos de diferentes instituições [Favela & Peña Mora, 2001] e, dependendo do nível de distribuição, também envolve diferenças culturais e de fuso-horário [Damian et al., 2006]. Tais dificuldades podem ser desestimulantes, entretanto uma possível solução para contornar tais dificuldades é a utilização da simulação do cenário de distribuição geográfica, como sugerido por [Keenan et al., 2010], o que possivelmente despenderá um menor esforço em comparação à utilização de distribuição real das equipes de desenvolvimento.

A estratégia de ensino proposta utiliza a simulação de um cenário DDS, em um projeto acadêmico, visando promover o desenvolvimento de habilidades de comunicação e colaboração de alunos participantes do projeto. Os passos desta estratégia são enumerados da seguinte forma:

1. Escolhe-se uma atividade pertencente ao ciclo de vida de software e um projeto a ser desenvolvido;
2. São criados os artefatos de entrada (e.g. documento de requisitos) a serem fornecidos aos alunos;
3. São inseridos problemas nos artefatos de entrada (e.g. ambiguidade de requisitos e omissão de informações)
4. Formam-se n equipes de alunos para atuarem no projeto;
5. São criados n personagens fictícios (avatars) que irão interagir, via internet, com as n equipes;

6. As equipes recebem os artefatos de entrada e devem interagir com os avatares de modo a sanar os problemas presentes nos artefatos de entrada e gerar os artefatos de saída (e.g. casos de uso).
7. Após a conclusão do projeto os alunos realizam uma auto-avaliação de suas habilidades de comunicação e colaboração.

A inserção de erros nos artefatos de entrada (passo 3) tem como objetivo forçar a comunicação, pois a equipe virtual “criou” tais artefatos e portanto saberá como ajudar a sanar as inconsistências. A criação de problemas como esses é sugerida por Dawson [2000], pois segundo este pesquisador tais problemas ocorrem em cenários reais de desenvolvimento. Os avatares criados, no passo 5, devem ser operados por um ou mais monitores cujas identidades serão mantidas em sigilo dos participantes. A interação com as equipes deve ocorrer, preferencialmente, de forma simultânea para que as diferentes equipes possam ter a sensação de que existem personalidades reais e distintas por trás dos avatares. A Figura 6.1 ilustra os passos da estratégia proposta.

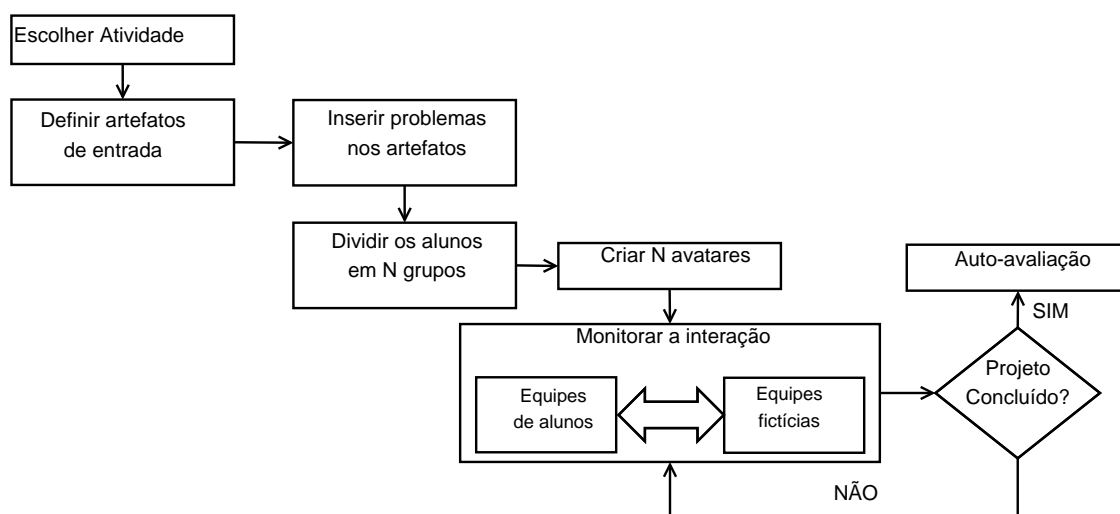


Figura 6.1. Representação gráfica da estratégia proposta

6.2 Estudos Experimentais

Nesta seção são apresentados os estudos experimentais realizados. O primeiro estudo realizado funcionou como um estudo preliminar, no qual buscou-se observar a dinâmica de interação entre equipes distribuídas, para deste modo ajustar os instrumentos de coleta e análise de dados. Neste estudo, utilizou-se distribuição real das equipes de desenvolvimento.

No segundo estudo buscou-se aperfeiçoar as ferramentas de coleta de dados e as atividades realizadas, com base nas lições aprendidas do experimento anterior. Após cada um dos experimentos foram realizadas análises qualitativas utilizando procedimentos de *Grounded Theory* [Strauss & Corbin, 1998].

Os estudos aqui relatados geraram duas publicações: uma com foco na estratégia de ensino em si [Fortaleza et al., 2013a] e outra com foco no aperfeiçoamento de habilidades [Fortaleza et al., 2013b]. A seguir descreve-se o planejamento e os métodos de coleta e análise utilizados.

6.2.1 Planejamento e Métodos de Coleta e Análise de Dados

Nos estudos apresentados fez-se uso de DDS em atividades de Engenharia de Requisitos: especificação e análise de requisitos. Buscou-se caracterizar a percepção dos alunos participantes com relação ao impacto da distância sobre os processos de comunicação e colaboração e também sua percepção sobre uma eventual aquisição ou aprimoramento de habilidades. A utilização de atividades relacionadas à Engenharia de Requisitos é justificada pelo fato de a fase de Engenharia de Requisitos ser considerada uma das mais críticas e pela mesma demandar maior interação por parte da equipe de desenvolvimento [Vale et al., 2011; Lemos & de Souza, 2008].

Em todos os estudos realizados a autorização da utilização dos dados para a pesquisa foi fornecida espontaneamente por cada aluno por meio de um Termo de Consentimento Livre e Esclarecido (TCLE), no qual foi explicada a natureza voluntária da participação nesta pesquisa de mestrado. O TCLE utilizado é apresentado no Apêndice B.

6.2.1.1 Pesquisas Qualitativas e *Grounded Theory*

Em Engenharia de Software, quando se considera o aspecto humano, existem diversas variáveis que não são quantitativas e para que se obtenha maior compreensão do fenômeno de interesse deve-se fazer uso de uma abordagem qualitativa [Seaman, 1999]. A pesquisa apresentada nesta dissertação utilizou procedimentos de uma técnica de análise qualitativa conhecida como *Grounded Theory* (GT) [Strauss & Corbin, 1998].

Grounded Theory ou Teoria Fundamentada em Dados é um método de análise qualitativa que tem por objetivo a criação de teorias sobre um determinado fenômeno de interesse [Strauss & Corbin, 1998]. Este método tem como base o processo de codificação.

A codificação consiste em identificar conceitos, ou códigos, bem como na criação de categorias, as quais representam conjunto de códigos com um alto grau de abstração.

De acordo com Strauss & Corbin [1998] o processo de codificação é composto por três fases:

- Codificação aberta: consiste na identificação de códigos e categorias iniciais, envolve a “quebra”, análise, comparação e conceitualização dos dados;
- Codificação axial: consiste na identificação de possíveis relações entre as diferentes categorias identificadas durante a codificação aberta;
- Codificação seletiva: corresponde ao refinamento da análise, que leva a identificação de uma categoria central, a qual agrupa as demais categorias

A Figura 6.2 ilustra o processo de codificação aberta.

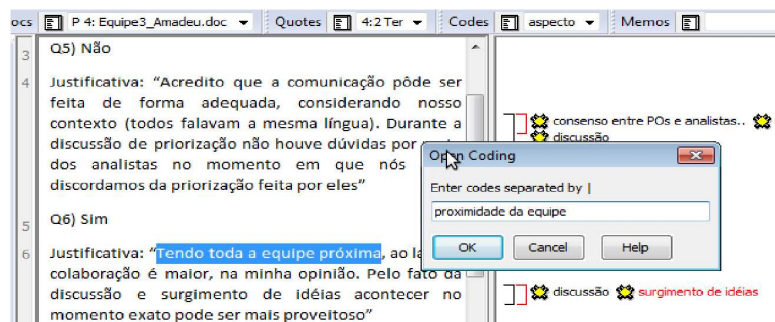


Figura 6.2. Exemplo de codificação aberta

Nesta pesquisa utilizou-se apenas codificação aberta e axial. Para realizar a codificação seletiva seria necessário atingir a saturação teórica do fenômeno estudado [Matos et al., 2013; Santos et al., 2012], além disso, o objetivo dos estudos experimentais realizados era entender como o uso de DDS pode afetar o processo de aperfeiçoamento de habilidades, e não a criação de uma teoria.

6.2.2 Estudo Experimental Preliminar

6.2.2.1 Descrição da amostra e da atividade realizada

Os participantes foram alunos de graduação e pós-graduação (tanto mestrado quanto doutorado) da Universidade Federal do Amazonas (UFAM), cursando a disciplina Tópicos Especiais em Engenharia de Software. Nesta disciplina um dos tópicos abordados foi Engenharia de Requisitos em projetos DDS. Os alunos eram provenientes dos seguintes cursos: Ciência da Computação, Engenharia da Computação, Mestrado em Informática e Doutorado em Informática. Nenhum dos alunos envolvidos no estudo possuía experiência anterior em DDS.

A turma foi dividida em grupos com quatro componentes em cada um, totalizando 12 participantes, entretanto no pós-experimento não foi possível obter o questionário de um dos alunos envolvidos, deste modo os dados considerados nos questionários pós-experimento são provenientes de 11 alunos.

Cada participante respondeu a um questionário de caracterização. Os grupos foram balanceados em função da experiência de cada membro em projetos reais de desenvolvimento de software. Foram criadas três categorias de experiência: baixa (menos de um ano de experiência), média (de um a dois anos de experiência) e alta (mais de dois anos de experiência); os alunos foram divididos nestas categorias e randomicamente foram escolhidos alunos de cada uma delas para formar as equipes. Cada grupo foi dividido em dois, formando sub-equipes.

As sub-equipes ficaram em localidades diferentes: uma sub-equipe ficou em sala de aula e a outra ficou em um laboratório localizado a 95 metros de distância (quando a distância é superior a 30 metros a comunicação é afetada da mesma forma que vários quilômetros de distância [Allen, 1977; Herbsleb & Mockus, 2003]).

Neste primeiro estudo, as equipes participantes fizeram uso de SCRUM. Decidiu-se utilizar esta metodologia ágil porque existem indícios na literatura de que a utilização de práticas ágeis tem impacto positivo sobre as habilidades de comunicação [Pikkarainen et al., 2008].

Em cada sub-equipe um representante foi escolhido, pelo pesquisador que estava conduzindo o experimento, para atuar como líder da sub-equipe. Os líderes ficaram responsáveis pela comunicação entre as duas sub-equipes. Deste modo apenas o líder da equipe de POs¹ poderia se comunicar com o líder da equipe de analistas. Para escolher o líder de cada equipe, o pesquisador responsável levou em consideração uma questão presente no questionário de caracterização que versava sobre a experiência com liderança e o “sentir-se bem” ao liderar.

A Figura 6.3 representa essa divisão das equipes. Nesta figura, as setas representam fluxos de comunicação. Em cada uma das localidades havia um monitor, no caso da sala de aula este monitor também exercia o papel de cliente. Os monitores eram pesquisadores.

Os participantes realizaram atividades de elicitação e priorização de requisitos, além de dividi-los em três *sprints*. Os alunos que ficaram em sala de aula receberam do “cliente” uma descrição verbal, face-a-face, de um sistema de software para a criação de um repositório online de cursos GSE. Cada sub-equipe teve de criar *user stories*, baseadas nessa descrição de sistema, e enviá-las através de e-mail para as sub-equipes

¹PO é a sigla para *Product Owner* que em SCRUM é o representante dos *stakeholders* junto a equipe de desenvolvedores

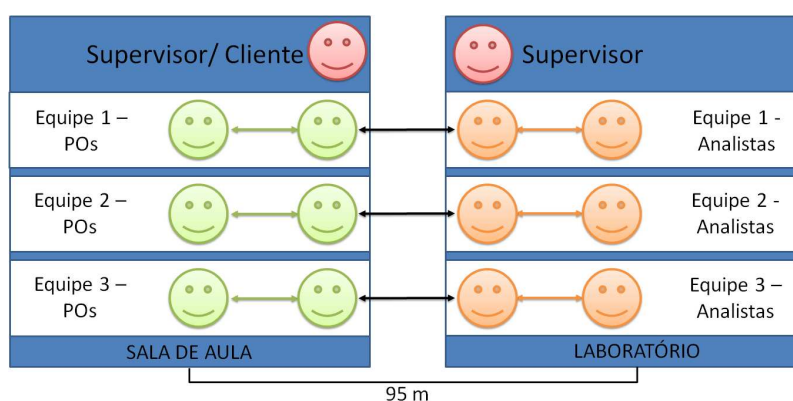


Figura 6.3. Configuração das equipes no Estudo Experimental Preliminar

que ficaram no laboratório.

Neste ponto do experimento, as sub-equipes de analistas tiveram de priorizar as *user stories*, separá-las em três *sprints*, e desenhar um *mockup* para uma *user story* de cada *sprint*. Entretanto, para atingir esse objetivo, ambas as sub-equipes (POs + analistas) tiveram de interagir através de *chat* e e-mail. Os registros de interação foram gravados e enviados aos pesquisadores através de e-mail para análise.

6.2.2.2 Resultados e Limitações

Apesar de a atividade ser aparentemente simples, houve relatos, registrados nos questionários pós-experimento de que a participação nesta atividade com DDS aperfeiçoou habilidades de comunicação e colaboração dos alunos envolvidos. No referido questionário havia duas questões sobre percepção do ganho de habilidades.

Optou-se por basear os dados na percepção dos alunos envolvidos porque as habilidades de comunicação e colaboração são muito subjetivas, portanto difíceis de quantificar, e utilizar percepção é a mesma abordagem utilizada por vários pesquisadores [Devlin & Phillips, 2010; Prikladnicki, 2011; Begel & Nagappan, 2008; Rivera-Ibarra et al., 2010]. As questões utilizadas para percepção foram as seguintes:

- Como você percebe o nível de influência da atividade desenvolvida sobre suas habilidades de comunicação?
- Como você percebe o nível de influência da atividade desenvolvida sobre suas habilidades de colaboração?

Para ambas as questões as alternativas do questionário pós-experimentos eram as seguintes:

- **Nenhum impacto:** “Acredito que não houve nenhum impacto da atividade realizada sobre minhas habilidades”;
- **Impacto leve:** “Acredito que a atividade serviu para aperfeiçoar um pouco minhas habilidades”;
- **Impacto alto:** “Acredito que não houve nenhum impacto da atividade realizada sobre minhas habilidades”.

Os resultados obtidos neste primeiro experimento indicaram que cerca de 54% dos alunos perceberam algum impacto da atividade sobre suas habilidades de comunicação. Já com relação às habilidades de colaboração, cerca de 90% dos participantes perceberam algum aperfeiçoamento, ou seja, escolheram a alternativa “impacto leve” ou “impacto alto”. Na Figura 6.4 é apresentado o resultado da percepção dos alunos com relação ao impacto da atividade realizada sobre suas habilidades de comunicação. Como é possível observar nesta figura, a maior parte dos alunos notou algum impacto sobre estas habilidades, principalmente nas de colaboração.

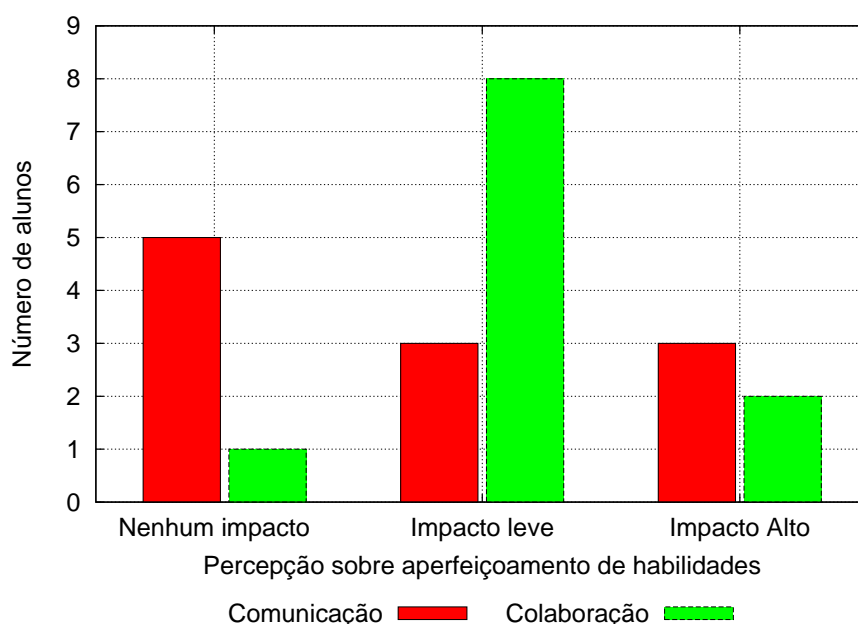


Figura 6.4. Resultados do estudo experimental preliminar

Apesar de ter-se obtido um resultado aparentemente positivo com este estudo, tais resultados possuem ameaças à validade, pois os alunos já se conheciam antes da realização do estudo. Entretanto este estudo preliminar foi útil aos seguintes propósitos: (i) fornecer maior compreensão sobre como se dá a comunicação entre alunos em um cenário de distribuição geográfica; e (ii) identificar oportunidades de melhoria

no questionário a ser utilizado no estudo de observação. Este novo questionário foi utilizado no estudo observacional que será descrito na seção seguinte.

Após a execução deste estudo preliminar, foram feitas as seguintes mudanças no questionário pós-experimento:

1. Deixou-se de perguntar explicitamente sobre ganhos em termos de habilidades de comunicação e colaboração, pois identificou-se que esta questão poderia induzir o participante;
2. Adicionou-se uma lista de possíveis ganhos trazidos pela atividade, e o participante deveria escolher uma ou mais opções. Dentre as opções estavam listadas as habilidades de comunicação e colaboração;
3. Foram adicionadas questões sobre satisfação e motivação;
4. Foram adicionadas mais “questões abertas”

6.2.3 Experimentação da Estratégia Proposta

6.2.3.1 Descrição da amostra e da atividade realizada

Este estudo foi realizado com alunos de graduação dos cursos de Sistemas de Informação e Ciência da Computação de outra instituição de ensino e teve a duração de um semestre acadêmico. Os alunos foram agrupados em cinco equipes e cada equipe teve de interagir com uma equipe virtual. As equipes foram formadas a partir do balanceamento da experiência profissional dos alunos envolvidos, seguindo procedimentos análogos aos adotados no estudo piloto. Os alunos não tomaram conhecimento da localização ou identidade do operador dos membros da equipe virtual.

O experimento realizado foi um estudo de observação [Shull et al., 2001]. O objetivo deste estudo experimental, estruturado de acordo com o paradigma GQM [Basili & Rombach, 1988] é apresentado na Tabela 6.1.

Os alunos receberam um documento de requisitos e tinham de especificar os casos de uso e montar os respectivos diagramas de casos de uso. O documento era referente a um sistema de gerenciamento de conferências científicas, este sistema foi escolhido pelo professor da turma.

O documento de requisitos fornecido continha propositalmente muitas ambiguidades e falta de informações, como sugerido nos “20 truques sujos” propostos por Dawson [2000]. Visando com isto forçar a interação dos alunos com a equipe virtual para solucionar dúvidas de entendimento. Esperava-se que com estas interações os participantes pudessem treinar suas habilidades de comunicação e colaboração.

Tabela 6.1. Objetivo do estudo de observação

Analisar	a percepção dos alunos, fornecida através de <i>logs</i> de interação e questionários
Com o propósito de Com relação	compreender ao aperfeiçoamento de habilidades de comunicação e colaboração; e, à satisfação com a atividade realizada
Do ponto de vista No contexto	dos pesquisadores de um experimento controlado com alunos que nunca haviam utilizado DDS

Os participantes foram supervisionados pelo professor da turma que, ao final do experimento, forneceu suas impressões através de uma entrevista semi-estruturada, cujo roteiro utilizado é apresentado no Apêndice C. Também foram coletados dados de *logs* de interação e questionários. Durante a análise dos resultados, buscou-se realizar a triangulação dos dados obtidos.

Este estudo teve a participação de 17 alunos, entretanto somente 14 forneceram os dados solicitados no questionário pós-experimento. Cada equipe possuía um líder que era responsável pela interação com o avatar da equipe virtual correspondente. As interações ocorreram via *chat* e e-mail.

O pesquisador que operou os avatares utilizou dois computadores, acessando cinco navegadores simultaneamente, o que lhe permitiu operar os cinco avatares ao mesmo tempo. Isto conferiu maior realismo ao experimento, pois os alunos estavam no mesmo laboratório e o operador pôde interagir com todas as equipes ao mesmo tempo. A Figura 6.5 ilustra a configuração do estudo observacional realizado. Já a Figura 6.6 ilustra uma interação.

Durante o semestre acadêmico, foram realizadas quatro reuniões com o professor da turma. Estas reuniões foram importantes para monitorar o desempenho dos alunos, pois forneceram uma visão sobre o que ocorreu no laboratório durante o experimento.

6.2.3.2 Resultados e limitações

No questionário pós-experimento, solicitou-se que os participantes escolhessem em uma lista em quais habilidades perceberam algum ganho. Dentre as opções estava a alternativa “habilidades de comunicação e colaboração”. Além desta lista, em uma questão aberta, os participantes deveriam responder de que forma perceberam este aperfeiçoamento de suas habilidades.

Dos 14 estudantes considerados neste estudo, cerca de 85% afirmaram ter perce-

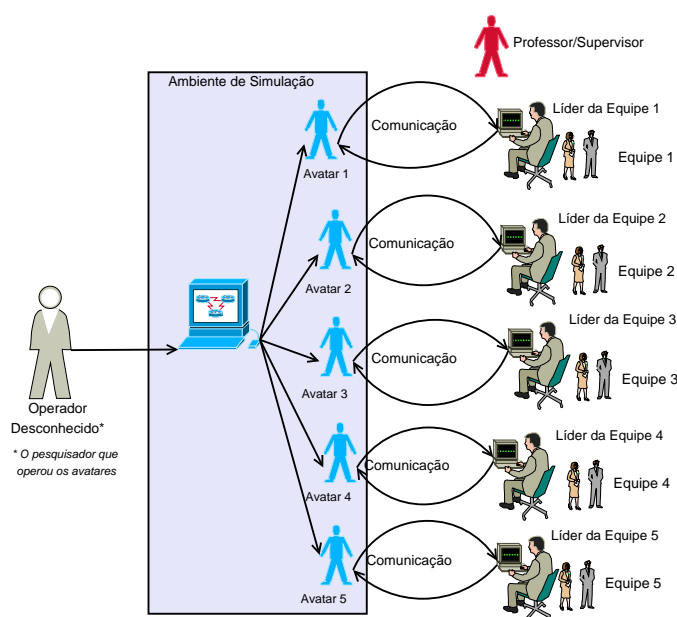


Figura 6.5. Configuração das equipes no Estudo de Observação

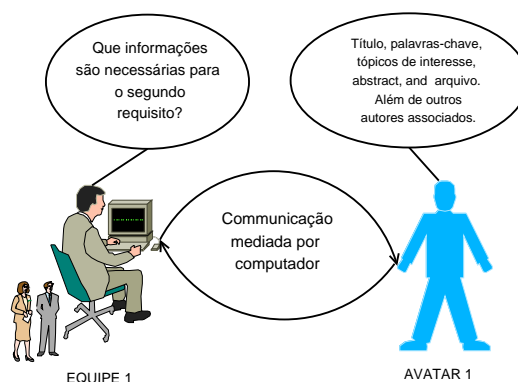


Figura 6.6. Exemplo de Interação no Estudo de Observação

bido aperfeiçoamento de suas habilidades de comunicação e colaboração. Os participantes também relataram benefícios relacionados a conhecimento técnico, capacidade de abstração, dentre outros. Na Figura 6.7 é apresentado um gráfico sobre o aperfeiçoamento de habilidades e aquisição de conhecimentos a partir da utilização da atividade com simulação do cenário DDS.

Os benefícios em relação às habilidades de comunicação e colaboração também são ilustrados pelos seguintes fragmentos retirados dos questionários:

*“A liderança de um projeto real me permitiu trabalhar a abstração e minhas habilidades de **colaboração**”* – Membro 1 da Equipe 3.

*“A atividade estimulou o relacionamento inter-pessoal, **fluxo de infor-***

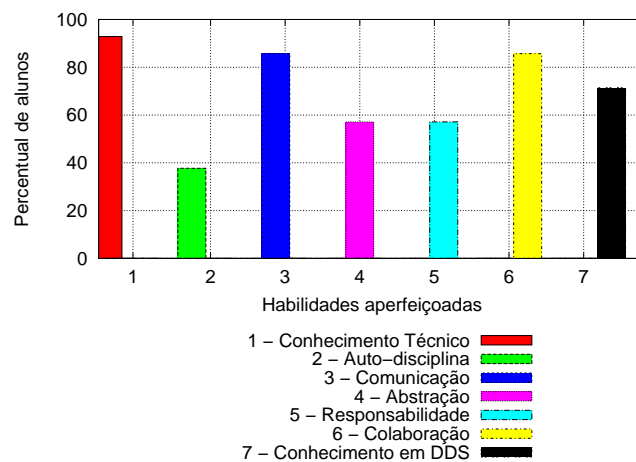


Figura 6.7. Resultados do Estudo de Observação

mação, abstração de cenários, colaboração...” – Membro 4 da Equipe 4.

Além disso, os estudantes afirmaram que conseguiram imaginar os processos de DDS e um cenário real de desenvolvimento, mesmo tratando-se de um cenário fictício, o que evidencia também um elevado grau de imersão dos participantes no ambiente simulado. Isto é evidenciado pelos seguintes fragmentos:

“A atividade me ajudou a entender como funciona o desenvolvimento de software desde o início” – Membro 1 da Equipe 1.

“Na minha opinião, a atividade trouxe a simulação de uma aplicação real, o que me levou, particularmente a utilizar protótipos para melhor compreensão do software” – Membro 2 da Equipe 2.

“A atividade me permitiu ver, de uma forma prática, como a modelagem acontece em um processo real de desenvolvimento” – Membro 3 da Equipe 2.

“Para mim, um aspecto positivo da atividade foi poder interagir com pessoas experientes” – Membro 2 da Equipe 1.

“Um aspecto positivo para mim foi a interação com um cliente real” – Membro 1 da Equipe 2.

É possível que a percepção de aperfeiçoamento de habilidades tenha ocorrido em função desta sensação de estar em ambiente real, o que reforça as evidências apontadas em outros trabalhos presentes na literatura [Dawson, 2000; Kaplan & Tan, 2010].

Neste estudo de observação, optou-se por coletar dados relativos à satisfação dos alunos envolvidos, pois de acordo com Kirkpatrick [1994] a satisfação é uma das dimensões da motivação. Além disso, existem pesquisadores que associam a motivação ao ganho de habilidades [Keller, 2009; Sharp et al., 2009]. Deste modo, visando capturar dados sobre satisfação, foram acrescentadas no questionário pós-experimento questões vindas da pesquisa conduzida por Savi et al. [2011], que por sua vez baseou-se no modelo ARCS [Keller, 2009]. Nestas questões, os participantes deveriam escolher alternativas em uma escala Likert de 5 pontos, circulando a opção desejada e, opcionalmente, tecendo comentários acerca da afirmativa correspondente.

As questões utilizadas são apresentadas na Tabela 6.2. Como é possível observar, as questões de número 5 e 6 também fornecem indícios para habilidades de comunicação e colaboração.

Tabela 6.2. Questões ligadas à satisfação e motivação - adaptada de Savi et al. [2011]

#	Afirmações	Avaliação					Comentários	
		Discordo Totalmente	-2	-1	0	+1		+2
1	Esta experiência contribuirá para minha vida profissional	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente
2	A experiência ajudou no meu aprendizado	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente
3	Esta abordagem com times distribuídos, foi mais útil ao aprendizado que as atividades co-allocadas	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente
4	Acredito que a atividade me propiciou uma experiência muito próxima de um cenário real de desenvolvimento	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente
5	Acredito que a atividade me levou a perceber a importância de transmitir informações corretas para chegar ao resultado final	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente
6	Acredito que a atividade me fez perceber a importância da colaboração para se atingir o resultado final	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente
7	Eu gostaria de participar de novas atividades com DDS porque foi eficiente para o aprendizado	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente

Os resultados de todos os participantes foram positivos, pois em todas estas questões, eles optaram pelas 3 alternativas superiores, tendendo ao “Concordo Totalmente”. Na Figura 6.8 tem-se uma representação deste resultado.

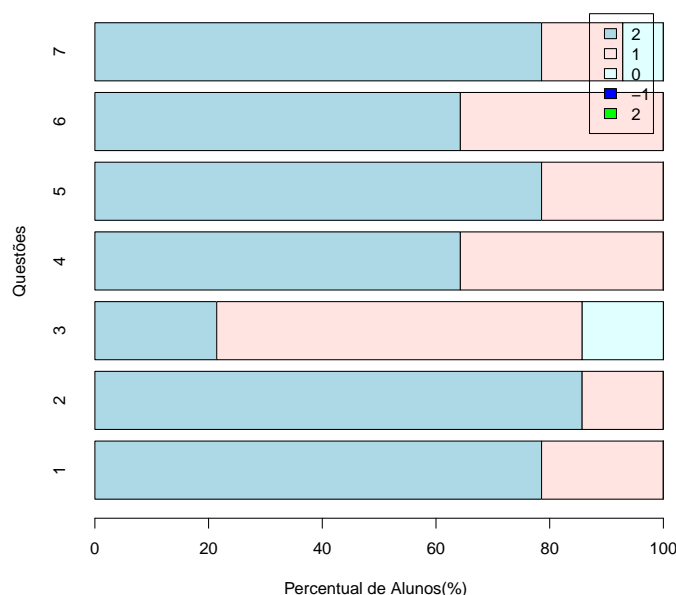


Figura 6.8. Representação gráfica das respostas dos participantes

Os questionários pós-experimento foram analisados utilizando os procedimentos de codificação aberta e axial do método GT. Os mesmos procedimentos foram utilizados para análise da transcrição da entrevista realizada com o professor da turma.

A partir do cruzamento dos *logs*, questionários e entrevista obtivemos uma lista de dificuldades, benefícios e sugestões de melhoria para a abordagem de ensino proposta. A Figura 6.9 apresenta a relação entre códigos identificados durante a codificação aberta, no que diz respeito às habilidades.

É importante notar que o professor citou durante a entrevista que os alunos estavam bastante motivados no início da atividade. Entretanto, nos questionários pós-experimento os alunos relatam que houve “falta de interesse da equipe local”. Esta aparente contradição pode ser explicada pelo fato de as equipes terem sido formadas levando em consideração somente a caracterização e desconsiderando as afinidades naturais entre os alunos da turma, o que foi feito objetivando simular o que acontece na indústria de desenvolvimento.

Deste modo, surge a hipótese de que para manter a motivação faz-se necessário considerar também estas afinidades, o que poderá ser investigado em trabalhos futuros. Na Figura 6.10 são apresentadas algumas afirmações feitas pelos envolvidos, ao longo do estudo experimental, que ilustram a motivação e desmotivação dos mesmos ao longo do projeto prático.

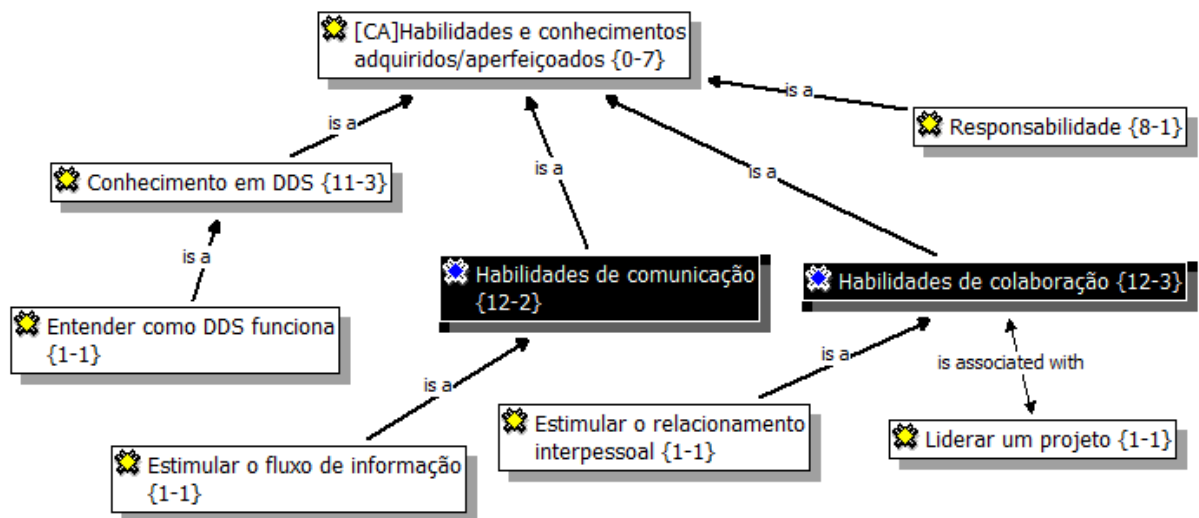


Figura 6.9. Representação do relacionamento entre os códigos identificados - Adaptado de Fortaleza et al. [2013b]

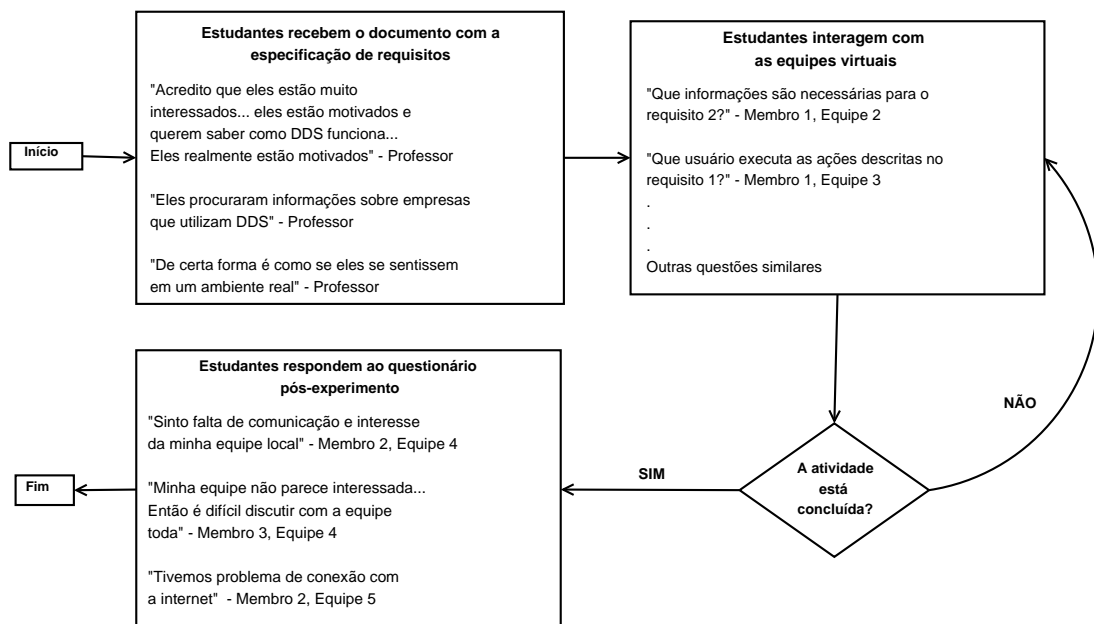


Figura 6.10. Motivação e desmotivação durante o estudo experimental - Adaptado de Fortaleza et al. [2013b]

Os resultados obtidos apontaram ganho em habilidades de comunicação e colaboração, o que reforça indícios prévios relatados na literatura [Begel & Nagappan, 2008; Prikladnicki, 2011]. Entretanto, os resultados apresentados foram obtidos a partir da utilização de um cenário no qual a distribuição geográfica foi simulada, isto pode indicar que é possível obter, através de simulação, resultados similares aos obtidos com a distribuição real. Todavia, para reforçar tais indícios faz-se necessária a condução

de um novo estudo no qual haja a comparação entre um cenário distribuído com um co-aloado, o que desponta como uma perspectiva de trabalho futuro.

Utilizar a estratégia de ensino proposta pode ser interessante pois requer menor esforço na implantação do que um curso com distribuição real das equipes de desenvolvimento. Além disso, trata-se de uma estratégia flexível, pois podem ser escolhidas outras atividades, diferentes critérios para formação das equipes, enfim, é possível adaptá-la para novos contextos.

Vale destacar que os resultados relatados correspondem a um contexto específico. Torna-se necessária a condução de novos estudos experimentais com diferentes instituições e atividades, de modo que em longo prazo se possa obter a generalização dos resultados.

Durante o experimento, tanto os alunos envolvidos quanto o professor da turma, perceberam algumas dificuldades. A Figura 6.11 é uma representação dos códigos identificados que foram relacionados a essas dificuldades. Se em futuras replicações, for traçado um plano de riscos para contornar estas dificuldades é possível que os benefícios trazidos pela estratégia proposta sejam ainda maiores.

6.3 Considerações Finais

Este capítulo apresentou a estratégia pedagógica proposta nesta pesquisa de mestrado, bem como os estudos experimentais conduzidos visando a obtenção de evidências sobre a eficiência da mesma para o aperfeiçoamento de habilidades de comunicação e colaboração. Também foram mostrados conceitos relacionados às pesquisas qualitativas e ao método *Grounded Theory*.

Os experimentos mostraram, em um contexto específico, que a utilização de Desenvolvimento Distribuído de Software é benéfica para o aperfeiçoamento de habilidades de comunicação e colaboração, e como mostrado no Capítulo 3 estas habilidades afetam um conjunto maior de habilidades. Portanto, ao se utilizar DDS em sala de aula é possível que se contribua para a formação de profissionais melhor capacitados.

Além disso, no estudo de observação conduzido, foram obtidos resultados positivos com a utilização da estratégia de ensino baseada em simulação. Portanto, é possível que utilizar a estratégia proposta seja uma alternativa viável para instituições que querem agregar o ensino de DDS a seus cursos regulares, mas que encontram dificuldade de estabelecer parcerias institucionais para essa empreitada. O capítulo seguinte apresenta as conclusões desta pesquisa de mestrado, uma síntese de suas contribuições e possibilidades de trabalhos futuros.

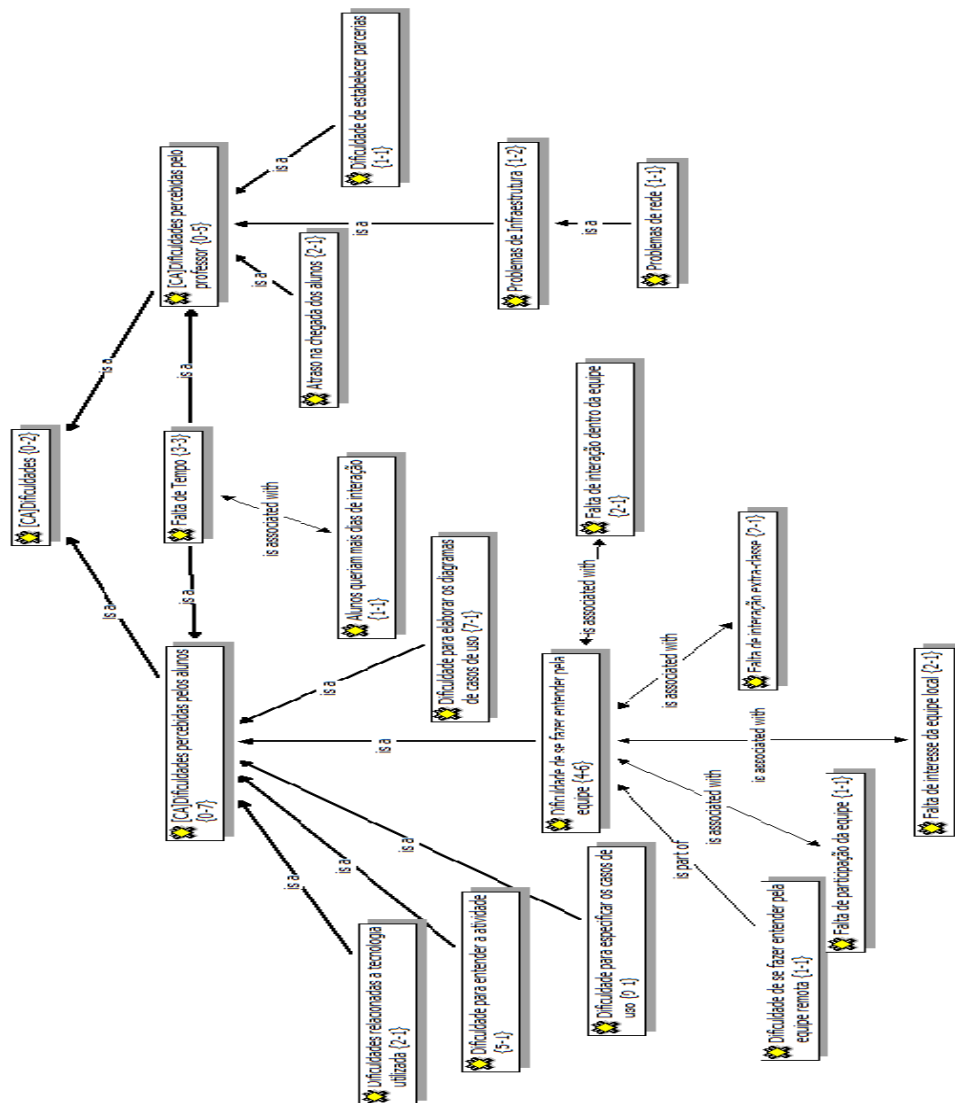


Figura 6.11. Representação das relações entre códigos relativos às dificuldades percebidas

Capítulo 7

Conclusão

Este capítulo encerra esta dissertação de mestrado, na qual uma estratégia para o desenvolvimento de habilidades de comunicação e colaboração em cursos de Engenharia de Software foi proposta. Deste modo, ao longo do capítulo será apresentada uma síntese dos resultados, contribuições e perspectivas de trabalhos futuros.

O restante deste capítulo encontra-se estruturado da seguinte forma: a Seção 7.1 apresenta o epílogo, a Seção 7.2 apresenta as contribuições geradas por esta pesquisa, a Seção 7.3 apresenta as limitações e, por fim, a Seção 7.4 lista algumas perspectivas de trabalhos futuros.

7.1 Epílogo

Este documento apresentou os resultados de uma pesquisa sobre a utilização de Desenvolvimento Distribuído de Software, em ambiente acadêmico, como meio de promover o desenvolvimento de habilidades de comunicação e colaboração. Durante esta pesquisa foi realizado um mapeamento das habilidades que são importantes para os engenheiros de software, um mapeamento de cursos de *Global Software Engineering* e a condução de dois estudos experimentais.

O foco desta pesquisa de mestrado foi buscar compreender como a utilização de DDS afeta o processo de aperfeiçoamento de habilidades de comunicação e colaboração, reforçando indícios presentes na literatura. Adicionalmente, avaliou-se a eficiência de se utilizar um cenário em que a distribuição geográfica das equipes é simulada com o mesmo objetivo.

Dada a natureza qualitativa da pesquisa conduzida, não é possível generalizar os resultados obtidos. Entretanto, esta pesquisa tem o mérito de contribuir para o

aumento do corpo de conhecimento sobre o tema investigado, o que em longo prazo poderá contribuir para a formulação de novas teorias. Espera-se que esta pesquisa leve outros pesquisadores a executarem novos estudos experimentais em diferentes contextos, aumentando deste modo o número de evidências e contribuindo para o processo científico, que em sua essência é iterativo.

7.2 Contribuições

7.2.1 Caracterização de Habilidades de Engenheiros de Software

Com base em um mapeamento sistemático da literatura, obteve-se uma lista das habilidades que são relatadas como importantes para o profissional de Engenharia de Software. Neste conjunto de habilidades percebeu-se que as habilidades de comunicação e colaboração afetam um conjunto maior de habilidades, e este foi o critério utilizado para delimitar habilidades de comunicação e colaboração como escopo desta pesquisa.

Este mapeamento de habilidades também demonstrou um crescimento do interesse da comunidade científica por este tema ao longo dos anos. Também ficou evidenciado que, em geral, pesquisas que versam sobre este tema utilizam a percepção do indivíduo como meio de avaliar a aquisição ou aperfeiçoamento de habilidades. Este mapeamento de habilidades gerou uma publicação [Fortaleza et al., 2011].

7.2.2 Caracterização de Cursos de GSE

Foi realizado um levantamento bibliográfico sobre cursos de *Global Software Engineering* (GSE), o qual permitiu a síntese das principais lições aprendidas reportadas por pesquisadores que fizeram uso desta modalidade de desenvolvimento em sala de aula. Observou-se que conseguir estabelecer parcerias institucionais e alinhar objetivos pedagógicos de diferentes instituições despende um grande esforço de planejamento, o que pode ser um fator impeditivo para pesquisadores que desejem propiciar a seus alunos a vivência de projetos geograficamente distribuídos.

Uma alternativa para contornar esta dificuldade é simular a distribuição geográfica, o que possivelmente requer menos esforço. Esta foi a linha adotada nesta pesquisa de mestrado.

O levantamento de cursos também evidenciou a necessidade da criação de um repositório de cursos GSE, visando à maior integração da comunidade científica. Isto

foi amplamente debatido pelos pesquisadores que participaram do *Collaborative Teaching of Globally Distributed Software Development - Community Building Workshop* de 2012, evento no qual os resultados obtidos pelo mapeamento de cursos encontram-se publicados [Fortaleza et al., 2012].

7.2.3 Estratégia para a Promoção do Desenvolvimento de Habilidades de Comunicação e Colaboração

A principal contribuição desta pesquisa é a estratégia para o desenvolvimento de habilidades de comunicação e colaboração em cursos de Engenharia de Software. Esta estratégia utiliza a simulação de um cenário de desenvolvimento geograficamente distribuído. Nos experimentos realizados, obteve-se resultados semelhantes aos obtidos quando se utiliza distribuição real.

A estratégia proposta é flexível e pode ser adaptada para outras fases do desenvolvimento de software, com um número variável de equipes a serem utilizadas, bem como variações da quantidade de operadores dos personagens fictícios a serem utilizados na simulação.

Deste modo, acredita-se que a estratégia proposta poderá ser útil às instituições que querem adotar DDS em seus cursos regulares de Engenharia de Software, mas que encontram dificuldades em utilizar distribuição real. A estratégia proposta gerou uma publicação [Fortaleza et al., 2013a].

7.2.4 Levantamento de Indícios sobre Aperfeiçoamento de Habilidades com o uso da Estratégia Proposta

Os estudos experimentais realizados mostraram, em um contexto específico, que a adoção da estratégia proposta em ambientes acadêmicos estimula o desenvolvimento de habilidades de comunicação e colaboração dos alunos, isto considerando-se como forma de avaliação a percepção dos mesmos. Com isto, resultados de trabalhos prévios reportados pela literatura foram reforçados.

Além disso, nos dois estudos realizados, os alunos também enfatizaram a aquisição de conhecimento técnico, o que é um indício de que apesar de se trabalhar habilidades de comunicação e colaboração, a abordagem utilizada não desconsiderou o aspecto técnico, o qual é indiscutivelmente importante para a formação de bons profissionais.

Vale ressaltar também que os participantes se mostraram motivados e demonstraram interesse de participar de novas atividades com utilização de DDS. Os resultados obtidos com relação à eficiência da utilização de DDS geraram uma publicação

[Fortaleza et al., 2013b], entretanto vale ressaltar que para resultados mais conclusivos faz-se necessária a replicação do estudo em um cenário no qual haja a comparação entre o uso de DDS e o uso de projetos co-allocados, o que poderá ser realizado em novas iniciativas de pesquisa.

7.3 Limitações

Esta pesquisa foi realizado tendo um número reduzido de participantes, o que não permite a generalização dos resultados obtidos. Por outro lado, pesquisas qualitativas visam a obtenção de um entendimento em profundidade de um tema estudado. Deste modo novas replicações fazem-se necessárias para que algum dia se possa generalizar os resultados obtidos. Além disso, os dados coletados foram baseados na percepção dos participantes e é possível que a utilização de testes psicológicos, ou algo semelhante, traga resultados mais objetivos.

A estratégia de ensino proposta foi avaliada somente com algumas atividades de Engenharia de Requisitos, é possível que utilizando-se um outro conjunto de atividades de outras fases do ciclo-de-vida do software, sejam obtidos outros resultados. Além disso, a utilização de avatares traz a necessidade de se ter uma pessoa os operando em tempo real, talvez a utilização de agentes virtuais programados possa contornar esta questão, todavia a utilização de um operador em tempo real pode ser vantajosa no sentido de se ter reações mais condizentes com um desenvolvedor real.

A utilização de DDS se mostrou eficiente neste contexto específico, entretanto não foi realizada uma comparação com atividades co-allocadas. Um novo experimento, com o intuito de efetuar este comparativo, pode ser visto como uma possibilidade de trabalho futuro.

7.4 Perspectivas de Trabalhos Futuros

As principais oportunidades de trabalhos futuros identificadas nesta dissertação são listadas a seguir:

- Utilização de outro conjunto de habilidades: nesta dissertação investigou-se a promoção de habilidades de comunicação e colaboração, entretanto existem outras habilidades que estão relacionados às atividades dos engenheiros de software (e.g. habilidades de negociação), portanto pode ser interessante do ponto de vista científico conduzir estudos que busquem promover o desenvolvimento de outro conjunto de habilidades;

- Utilização de outras atividades do ciclo de vida de software (e.g. atividades de V&V): nesta pesquisa a estratégia proposta fez uso de atividades de Engenharia de Requisitos, entretanto a Engenharia de Software envolve outras atividades, de modo que como trabalho futuro pode-se utilizar outras atividades para comparação com os resultados desta pesquisa identificando-se assim qual a influência do tipo de atividade executada para a eficiência da técnica proposta;
- Utilização de agentes virtuais programados para expressar emoções, no cenário simulado: a estratégia proposta utiliza avatares operados em tempo real por um ou mais monitores (no caso dos experimentos, um dos pesquisadores), uma possibilidade de pesquisa futura é a utilização de agentes virtuais programados, entretanto é possível que se haja perdas em termos de qualidade da interação. Um estudo comparativo seria interessante.
- Execução de um experimento em que se utilizem também projetos co-locados: deste modo será possível ver o real impacto da utilização da distribuição geográfica para o aperfeiçoamento de habilidades

Portanto a estratégia proposta, além de suas contribuições próprias, também serviu para a identificação de novas frentes de pesquisa a serem realizadas por parte da comunidade científica de Engenharia de Software.

Referências Bibliográficas

- Agerfalk, P. J.; Fitzgerald, B.; Holmström, H.; Lings, B.; Lundel, B. & Conchúir, E. O. (2005). A framework for considering opportunities and threats in distributed software development. In *Proceedings of the International Workshop on Distributed Software*, pp. 47–61, Paris, France. Austrian Computer Society.
- Akgün, A. E.; Keskin, H.; Byrne, J. & Imamoglu, S. Z. (2007). Antecedents and consequences of team potency in software development projects. *Information and Management*, 44(7):646–656.
- Allen, T. J. (1977). *Managing the Flow of Technology*. MIT Press.
- Almeida, E.; Dali, L.; Faulk, S.; Lima, C.; Rui, Z.; Weiss, D.; Ying, J.; Young, M. & Yu, L. (2012). Teaching Globally Distributed Software Development: An Experience Report. In *Proceedings of the 25th Conference on Software Engineering Education and Training, CSEE&T '12*, pp. 105–109, Nanjing, China. IEEE Computer Society.
- Asendorpf, J. B. (2004). *Psychologie der Persönlichkeit*. Springer.
- Atkinson, R. L.; Atkinson, R. C.; Smith, E. E. & Bem, D. J. (1993). *Introduction to Psychology*. Harcourt Brace Jovanovich, 11th edição.
- Audy, J.; Evaristo, R. & Watson-Manheim, M. B. (2004). Distributed Analysis: The Last Frontier? In *Proceedings of the 37th Hawaii International Conference on System Sciences*, Big Island, USA. IEEE Computer Society.
- Audy, J. & Prikladnicki, R. (2008). *Desenvolvimento Distribuído de Software: desenvolvimento de software com equipes distribuídas*. Elsevier, Rio de Janeiro.
- Bandura, A. (1986). *Social Foundations of Thought and Action: A Social Cognitive Theory*. Prentice Hall, New Jersey.
- Basili, V. R. & Rombach, H. D. (1988). The TAME Project: towards improvement-oriented software environments. *Transactions on Software Engineering*, 14(6):758–773.
- Beecham, S.; Baddoo, N.; Hall, T.; Robinson, H. & Sharp, H. (2008). Motivation in Software Engineering: A systematic literature review. *Information and Software Technology*, 50(9-10):860–878.
- Begel, A. & Nagappan, N. (2008). Global software development: who does it? In *Proceedings of the 3rd International Conference on Global Software Engineering*, pp. 195–199, Bangalore, India.
- Begel, A. & Simon, B. (2008). Struggles of new college graduates in their first software development job. In *Proceedings of the 39th Technical Symposium on Computer Science Education*, pp. 226–230, Portland, USA.
- Bellur, U. (2006). An Academic Perspective on Globalization in the Software Industry. In *Proceedings of the 30th Annual International Computer Software and Applications Conference, COMPSAC '06*, pp. 53–54, Chicago, USA. IEEE Computer Society.

- Biscaro, A. W. (1994). Métodos e técnicas em T&D. In Boog, G. G., editor, *Manual de Treinamento e Desenvolvimento ABTD*. Makron Books, São Paulo, 2nd edição.
- Bosnić, I.; Cavarac, I.; Orlic, M.; Žagar, M. & Crnković, I. (2011). Student Motivation in Distributed Software Development Projects. In *Proceedings of the 33rd International Conference on Software Engineering. In: Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development, CTGDSD '11*, pp. 31–35, Honolulu, USA.
- Callele, D. & Makaroff, D. (2006). Teaching requirements engineering to an unsuspecting audience. In *Proceedings of the 37th Technical Symposium on Computer Science Education*, pp. 433–437, Houston, USA.
- Capretz, L. F. & Ahmed, F. (2010). Why do we need personality diversity in software engineering? *SIGSOFT Software Engineering Notes*, 35(2):1–11.
- Carmel, E. & Prikladnicki, R. (2010). Does Time Zone Proximity Matter for Brazil? A Study of the Brazilian I.T. Industry. Technical report.
- Carver, J.; Nagappan, N. & Page, A. (2008). The Impact of Educational Background on the Effectiveness of Requirements Inspections: An Empirical Study. *IEEE Transactions on Software Engineering*, 34(6):800–812.
- Castro, T.; Castro, A. & Fuks, H. (2008). Aprendendo a Programar em Grupo. In *Anais do V Simpósio Brasileiro de Sistemas Colaborativos, SBSC '08*, pp. 45–54, Vila Velha, Brasil. IEEE Computer Society.
- Catania, J. T. (2006). An interdisciplinary practical approach to teaching the software development life-cycle. In *Proceedings of the 7th Information Technology Education Conference*, pp. 3–8, Minneapolis, USA.
- Cherry, S. & Robillard, P. N. (2008). The social side of software engineering-A real ad hoc collaboration network. *International Journal of Human-Computer Studies*, 66(7):495–505.
- Conte, T. U.; Maia, N.; Marques, A. B.; Mendes, E. & Travassos, G. H. (2011). Estudo sobre a Influência do Tipo de Personalidade do Inspetor no Desempenho de Inspeções de Usabilidade. In *Proceedings of the XIV Ibero-American Conference on Software Engineering*, pp. 211–224, Rio de Janeiro, Brazil.
- Crnković, I.; Bosnić, I. & Žagar, M. (2012). Ten tips to succeed in global software engineering education. In *Proceedings of the 34th International Conference on Software Engineering*, pp. 1225–1234, Zurich, Switzerland.
- Crnković, I.; Cavarac, I.; Fredriksson, J.; Land, R.; Žagar, M. & Akerholm, M. (2003). On the Teaching of Distributed Software Development. In *Proceedings of the 25th International Conference on Information Technology Interfaces*, pp. 237–242, Cavtat, Croatia.
- Cruz, S. S. J. O.; da Silva, F. Q. B.; Monteiro, C. V. F.; Santos, P. & Rossilei, I. (2011). Personality in Software Engineering: preliminary findings from a systematic literature review. In *Proceedings of the 15th Annual Conference on Evaluation & Assessment in Software Engineering, EASE '11*, pp. 1–10, Durham, UK.
- Curtis, B.; Hefley, W. E. & Miller, S. (1995). Overview of the People Capability Maturity Model. Technical report, Software Engineering Institute, Carnegie Mellon University, CMU/SEI-95-MM-01, Pittsburgh, USA.
- da Silva, F. Q. B. & França, A. C. C. (2012). Towards understanding the underlying structure of motivational factors for software engineers to guide the definition of motivational programs. *Journal of Systems and Software*, 85(2):216–226.

- Damian, D.; Hadwin, A. & Al-Ani, B. (2006). Instructional design and assessment strategies for teaching global software development: a framework. In *Proceedings of the 28th international conference on Software engineering*, ICSE '06, pp. 685–690, Shanghai, China. ACM.
- Damian, D. E. & Zowghi, D. (2002). The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In *Proceedings of the IEEE Joint International Conference on Requirements Engineering*, pp. 319–328, Essen, Germany. IEEE Computer Society.
- Dawson, R. (2000). Twenty dirty tricks to train software engineers. In *Proceedings of the 22nd International Conference on Software Engineering*, ICSE '00, pp. 209–218, Limerick, Ireland.
- de Souza, C. R. B.; Sharp, H.; Singer, J.; Cheng, L.-T. & Venolia, G. (2009). Guest Editors' Introduction: Cooperative and Human Aspects of Software Engineering. *IEEE Software*, 26(6):17–19.
- Devlin, M. & Phillips, C. (2010). Assessing competency in undergraduate software engineering teams. In *2010 IEEE Education Engineering Conference*, pp. 271–278, Madrid, Spain.
- Downey, J. (2005). A framework to elicit the skills needed for software development. In *Proceedings of the ACM SIGMIS CPR conference on Computer personnel research*, SIGMIS CPR '05, pp. 122–127, Atlanta, USA. ACM.
- Dybå, T. (2003). Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context. In *Proceedings of the 9th European Software Engineering Conference held jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, Helsinki, Finland.
- Favela, J. & Peña Mora, F. (2001). An experience in collaborative software engineering education. *IEEE Software*, 18(2):47–53.
- Feldt, R.; Angelis, L.; Torkar, R. & Samuelsson, M. (2010). Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*, 52(6):611–624.
- Ferreira, P. G. & da Silva, F. Q. B. (2008). Fatores humanos que influenciam a utilização de processos de software. In *Anais do VII Simpósio Brasileiro de Qualidade de Software*, SBQS '2008, Florianópolis, Brasil.
- Fortaleza, L. L.; Conte, T.; Marczak, S. & Prikladnicki, R. (2012). Towards a GSE International Teaching Network: Mapping Global Software Engineering Courses. In *ICSE 2012. In: Proceedings of the 2nd Collaborative Teaching of Globally Distributed Software Development*, pp. 1–5, Zurich, Switzerland.
- Fortaleza, L. L.; Matos Júnior, O. O.; Conte, T.; Vieira, S. R. C. & Prikladnicki, R. (2013a). Improving Software Engineers' Skills through the Simulation of Distributed Software Development in Academic Environments. In *Proceedings of the 25th International Conference on Software Engineering and Knowledge Engineering*, SEKE '13, Boston, USA.
- Fortaleza, L. L.; Prikladnicki, R. & Conte, T. U. (2011). Habilidades de Engenheiros de Software: uma análise qualitativa a partir de uma Revisão Sistemática. In *Anais do X Simpósio Brasileiro de Qualidade de Software: VII Workshop Olhar Sociotécnico Sobre a Engenharia de Software*, WOSES '11, Curitiba, Brasil.
- Fortaleza, L. L.; Vieira, S. R. C.; Matos Júnior, O. O.; Prikladnicki, R. & Conte, T. (2013b). Using distributed software development in improvement of communication and collaboration skills in SE courses: an observational study. In *Proceedings of the 26th Conference on Software Engineering Education and Training*, CSEE&T '13, San Francisco, USA.
- França, A. C. C. & da Silva, F. Q. B. (2010). Designing Motivation Strategies for Software Engineering Teams: an Empirical Study. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '2010, pp. 84–91, Cape Town, South Africa.

- Gotel, O.; Kulkarni, V.; Scharff, C. & Neak, L. (2008). Working across borders: overcoming culturally-based technology challenges in student global software development. In *Proceedings of the 21st Conference on Software Engineering Education and Training, CSEE&T '08*, pp. 33–40, Charleston, USA. IEEE Computer Society.
- Guinan, P. J.; Coopriider, J. G. & Faraj, S. (1998). Enabling software development team performance during requirements definition: a behavioral versus technical approach. *Information Systems Research*, 9(2):101–125.
- Hazzan, O. & Tomayko, J. (2004). Human Aspects of Software Engineering: The Case of Extreme Programming. In *Proceedings of the 5th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP '04*, pp. 303–311, Garmisch-Partenkirchen, Germany.
- Herbsleb, J. D. & Mockus, A. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29(6):481–494.
- Hogan, J. M. & Thomas, R. (2005). Developing the software engineering team. In *Proceedings of the 7th Australasian Conference on Computing Education, ACE '05*, pp. 203–210, New Castle, Australia. IEEE Computer Society.
- Jalali, S. & Wohlin, C. (2012). Systematic Literature Studies: Database Searches vs. Backward Snowballing. In *Proceedings of the 6th International Symposium on Empirical Software Engineering and Measurement, ESEM '12*, pp. 29–38, Lund, Sweden.
- Jazayeri, M. (2004). The education of a software engineer. In *Proceedings of the 19th International Conference on Automated Software Engineering*, pp. 18–27, Linz, Austria. IEEE Computer Society.
- Jiménez, M.; Piattini, M. & Vizcaíno, A. (2009). Challenges and improvements in distributed software development: a systematic review. *Advances in Software Engineering*, 2009:1–15.
- Kaplan, R. M. & Tan, J. (2010). Real world experiences in a software engineering course. In *Proceedings of the 2010 ACM Conference on Information Technology Education, SIGITE '10*, pp. 89–94, Midland, USA.
- Keenan, E.; Steele, A. & Jia, X. (2010). Simulating global software development in a course environment. In *Proceedings of the 5th International Conference on Global Software Engineering, ICGSE '10*, pp. 201–205, Princeton, USA. IEEE Computer Society.
- Keirse, D. & Bates, M. (1984). *Please Understand Me: Character and Temperament Types*. B & D Books, 5th edição.
- Keller, J. M. (2009). *Motivational Design for Learning and Performance: The ARCS Model Approach*. Springer.
- Kirkpatrick, D. L. (1994). *Evaluating Training Programs - The Four Levels*. Berrett-Koehler Publishers.
- Kitchenham, B.; Budgen, D. & Brereton, O. P. (2011). Using mapping studies as the basis for further research - A participant-observer case study. *Information and Software Technology*, 53(6):638–651.
- Kitchenham, B. & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report, Keele University and Durham University.
- Largent, D. L. & Lüer, C. (2010). "You mean we have to work together?!": A study of the formation and interaction of programming teams in a college course setting. In *Proceedings of the International Computing Education Research Workshop*, pp. 41–49, Aarhus, Denmark.

- Lemos, A. D. M. & de Souza, C. R. B. (2008). Desafios da Gerência de Conhecimento no Desenvolvimento de Software: Resultados de um Estudo Etnográfico. In *Anais do V Simpósio Brasileiro de Sistemas Colaborativos*, pp. 77–87, Vila Velha, Brazil.
- Lethbridge, T. C. (2000). What knowledge is important to a software professional? *Computer*, 33(5):44–50.
- Li, Y.; Chang, K.-C.; Chen, H.-G. & Jiang, J. J. (2010). Software development team flexibility antecedents. *Journal of Systems and Software*, 83(10):1726–1734.
- Linhares, G. B. a. R.; Borges, M. R. S. & Antunes, P. (2008). Negociação-Colaboração nas Revisões Técnicas Formais de Especificações Funcionais. In *Anais do V Simpósio Brasileiro de Sistemas Colaborativos*, SBSC '08, pp. 122–133, Vila Velha, Brasil.
- Martínez, L. G.; Licea, G.; Rodríguez-Díaz, A. & Castro, J. R. (2010). Experiences in software engineering courses using psychometrics with RAMSET. In *Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '10, pp. 244–248, Bilkent, Ankara, Turkey. ACM.
- Matos, O.; Fortaleza, L.; Conte, T. & Mendes, E. (2013). Realising web effort estimation: a qualitative investigation. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, EASE '13, pp. 12–23, Porto de Galinha, Brazil.
- Matos Júnior, O. O.; Secatti, V. P.; Santos, D. V.; Oliveira, H. A. B. F. & Conte, T. U. (2010). Aplicando Grounded Theory para Compreender os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processo de Software. In *Anais do IX Simposio Brasileiro de Qualidade de Software: VI Workshop Olhar Sociotécnico Sobre a Engenharia de Software*, WOSES '10, Belém, Brasil.
- McConnell, J. V. (1978). *Psicologia*. Iteramericana, Rio de Janeiro, Brasil, 2 edição.
- McCrae, R. R. & Costa Jr, P. T. (2003). *Personality in adulthood - a Five-Factor Theory Perspective*. Guilford Press, New York, USA, 2nd edição.
- Meira, A. F. (2009). *Um estudo sobre as relações entre necessidades funcionais e comportamentais para a área de Garantia da Qualidade de Software em empresas de desenvolvimento de software*. Dissertação de mestrado, Universidade Federal de Pernambuco.
- Meira, A. F. & da Silva, F. Q. B. (2009). Habilities and Behavioural Profiles of SQA Professionals Related to Process Maturity Levels. In *Anais do XXIII Simpósio Brasileiro de Engenharia de Software*, SBES '2009, pp. 136–144, Fortaleza, Brasil.
- Mouly, G. J. (1973). *Psicologia Educacional*. Pioneira, São Paulo, Brasil.
- Mullick, N.; Bass, M.; El Houda, Z.; Paulish, D. J.; Cataldo, M.; Herbsleb, J. D.; Bass, L. & Sangwan, R. (2006). Siemens Global Studio Project: Experiences Adopting an Integrated GSD Infrastructure. In *Proceedings of the International Conference on Global Software Engineering*, ICGSE '06, pp. 203–212, Florianópolis, Brazil. IEEE Computer Society.
- Paixão, C. C.; Fortaleza, L. L. & Conte, T. U. (2012). Um Estudo Preliminar sobre as Implicações de Tipos de Personalidade no Ensino de Computação. In *Anais do XX Workshop sobre Educação em Informática*, WEI '12, Curitiba, Brasil.
- Penna, A. G. (2001). *Introdução a motivação e emoção*. LTC, Rio de Janeiro.
- Petersen, K.; Feldt, R.; Mujtaba, S. & Mattsson, M. (2008). Systematic Mapping Studies in Software Engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE '08, pp. 71–80, Bari, Italy.

- Pieterse, V.; Kourie, D. G. & Sonnekus, I. P. (2006). Software engineering team diversity and performance. In *Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, volume 204, pp. 180–186, Somerset West, South Africa.
- Pikkarainen, M.; Haikara, J.; Salo, O.; Abrahamsson, P. & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3):303–337.
- Prikladnicki, R. (2009). Exploring Propinquity in Global Software Engineering. In *Proceedings of the 4th International on Global Software Engineering, ICGSE '09*, pp. 133–142, Limerick, Ireland. IEEE Computer Society.
- Prikladnicki, R. (2011). Can distributed software development help the practitioners to become better software engineers?: insights from academia. In *Proceedings of the 33rd International Conference on Software Engineering. In: Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development, CTGDSD '11, Waikiki, Honolulu, Hawaii*.
- Prikladnicki, R. & Audy, J. L. N. (2010). Process models in the practice of distributed software development: A systematic review of the literature. *Information and Software Technology*, 52(8):779–791.
- Prikladnicki, R.; Marczak, S.; Conte, T. U.; de Souza, C.; Audy, J. L. N.; Marques, A. B. & Orsoletta, R. A. D. (2011). The Evolution and Impact of the Research in Distributed Software Development in Brazil. In *Anais do XXV Simpósio Brasileiro de Engenharia de Software, SBES '11*, pp. 126–131, São Paulo, Brasil. IEEE Computer Society.
- Purvis, M.; Purvis, M. & Cranefield, S. (2004). Educational experiences from a Global Software Engineering (GSE) project. In *Proceedings of The 6th Conference on Australasian Computing Education, ACE '04*, pp. 269–275, Darlinghurst, Australia. Australian Computer Society.
- Richardson, I.; Milewski, A. E.; Keil, P. & Mullick, N. (2006). Distributed Development â an Education Perspective on the Global Studio Project. In *Proceedings of the 28th International Conference on Software Engineering, ICSE '06*, pp. 679–684, Shangai, China. ACM.
- Richardson, I.; Paulish, D.; Moore, S.; Casey, V. & Zage, D. (2007). Globalizing Software Development in the Local Classroom. In *Proceedings of the 20th Conference on Software Engineering Education & Training, CSEE&T '07*, pp. 64–71, Dublin, Ireland. IEEE Computer Society.
- Rivera-Ibarra, J. G.; Rodriguez-Jacobo, J.; Fernández-Zepeda, J. A. & Serrano-Vargas, M. A. (2010). Competency framework for software engineers. In *Proceedings of the 23rd Conference on Software Engineering Education and Training*, pp. 33–40. IEEE Computer Society.
- Robbins, S. P. (2005). *Comportamento Organizacional*. Prentice-Hall, São Paulo.
- Robillard, P. N. & Robillard, M. P. (2000). Types of collaborative work in software engineering. *Journal of Systems and Software*, 53(3):219–224.
- Ruff, S. & Carter, M. (2009). Communication learning outcomes from software engineering professionals: A basis for teaching communication in the engineering curriculum. In *Proceedings of the 39th Frontiers in Education Conference*, pp. 1271–1276, San Antonio, TX, USA. IEEE Computer Society.
- Santos, D. V.; de Souza, C. R. B.; Vilela, D.; Santos, G.; Prikladnicki, R. & Conte, T. U. (2012). The Influence of Human Aspects on Software Process Improvement: Qualitative Research Findings and Comparison to Previous Studies. In *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering, EASE '12*, pp. 121–125, Ciudad Real, Spain.

- Santos, D. V.; Vilela Júnior, D. C.; de Souza, C. & Conte, T. U. (2011). Aspectos humanos que afetam um programa de melhoria de processo de software - Uma análise qualitativa. In *Proceedings of the XIV Ibero-American Conference on Software Engineering, CibSE '11*, pp. 157–170, Rio de Janeiro, Brazil.
- Savi, R.; von Wangenheim, C. G. & Borgatto, A. F. (2011). A Model for the Evaluation of Educational Games for Teaching Software Engineering. In *Anais do XXV Simpósio Brasileiro de Engenharia de Software, SBES '11*, pp. 194–203, São Paulo, Brasil.
- Seaman, C. B. (1999). Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572.
- Sharp, H.; Baddoo, N.; Beecham, S.; Hall, T. & Robinson, H. (2009). Models of motivation in software engineering. *Information and Software Technology*, 51(1):219–233.
- Shull, F.; Carver, J. & Travassos, G. H. (2001). An empirical methodology for introducing software processes. In *Proceedings of the 8th European Software Engineering*, pp. 288–296, Vienna, Austria.
- Sjøberg, D. I. K.; Dybå, T. & Jørgensen, M. (2007). The Future of Empirical Methods in Software Engineering Research. In *XXIX International Conference on Software Engineering. In: Workshop on Future of Software Engineering, FOSE '07*, pp. 358–378, Minneapolis, MN, USA. IEEE Computer Society.
- Stillier, E. & LeBlanc, C. (2002). Effective software engineering pedagogy. *Journal of Computing Sciences in Colleges*, 17(6):124–134.
- Strauss, A. C. & Corbin, J. M. (1998). *Basics of qualitative research: techniques and procedures for developing Grounded Theory*. SAGE, 3rd edição.
- Surakka, S. (2007). What subjects and skills are important for software developers? *Communications of the ACM*, 50(1):73–78.
- Swigger, K.; Brazile, R.; Harrington, B.; Peng, X. & Alpaslan, F. (2006). Teaching Students How to Work in Global Software Development Environments. In *Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 1–7, Atlanta, USA.
- Teague, J. (1998). Personality type, career preference and implications for computer science recruitment and teaching. In *Proceedings of the 3rd Australasian Conference on Computer Science Education*, pp. 155–163, Brisbane, Queensland, Australia. ACM.
- Turley, R. T. & Bieman, J. M. (1995). Competencies of exceptional and non exceptional software engineers. *Journal of Systems and Software*, 28(1):19–38.
- Uris, A. (1966). *Formação de dirigentes: idéias, pessoas e instituições que criam e prejudicam a administração*. Editora Ibrasa.
- Vale, L.; Albuquerque, A. B. & Beserra, P. (2011). A Importância da Qualidade Profissional dos Analistas de Requisitos para o Sucesso dos Projetos de Software: um Estudo para Identificar as Habilidades Mais Relevantes. In *Anais do XXV Simpósio Brasileiro de Engenharia de Software*, São Paulo, Brasil.
- Walker, E. L. & Slotterbeck, O. A. (2002). Incorporating realistic teamwork into a small college software engineering curriculum. *Journal of Computing Sciences in Colleges*, 17(6):115–123.
- Webster, J. & Watson, R. T. (2002). Analyzing the past to prepare for the future: writing a literature review. *MIS Quarterly*, 26(2):xiii – xxiii.
- Whitehead, J. (2007). Collaboration in Software Engineering: A Roadmap. In *Proceedings of the Future of Software Engineering, FOSE '07*, pp. 214–225, Minneapolis, USA. IEEE Computer Society.

Apêndice A

Lista de artigos selecionados no mapeamento sistemático de habilidades

Os procedimentos adotados para a seleção destes artigos estão descritos no Capítulo 3.

1. Abrahamsson, P., & Kautz, K. (2002). The personal software process: experiences from Denmark. *Proceedings of the 28th Euromicro Conference* (pp. 367 - 374). Dortmund, Germany: IEEE Computer Society
2. Akgün, A. E., Keskin, H., Byrne, J., & Imamoglu, S. Z. (2007). Antecedents and consequences of team potency in software development projects. *Information and Management*, 44(7), 646-656.
3. Al-Khatib, W. G., Bukhres, O., & Douglas, P. (1995). An empirical study of skills assessment for software practitioners. *Information Sciences - Applications*, 4(2), 83-118.
4. Avison, D., & Wilson, D. (2001). A viewpoint on software engineering and information systems: What we can learn from the construction industry? *Information and Software Technology*, 43(13), 795-799.
5. Bareiss, R., & Mercier, G. (2010). A Graduate Education in Software Management and the Software Business for Mid-Career Professionals. *Proceedings of the 23rd IEEE Conference on Software Engineering Education and Training* (pp. 65-72).
6. Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in Software Engineering: A systematic literature review. *Information and Software Technology*, 50(9-10), 860-878.
7. Begel, A., & Simon, B. (2008). Struggles of new college graduates in their first software development job. *Proceedings of the 39th Technical Symposium on Computer Science Education* (pp. 226-230).
8. Beranek, G., Zuser, W., & Grechenig, T. (2005). Functional group roles in software engineering teams. *Proceedings of the International Conference on Software Engineering*. In: *Workshop on Human and Social Factors of Software Engineering* (Vol. 30, pp. 1-7)
9. Boehm, B. (2000). The art of expectations management. *Computer*, 33(1), 122-124.
10. Callele, D., & Makaroff, D. (2006). Teaching requirements engineering to an unsuspecting audience. *Proceedings of the 37th Technical Symposium on Computer Science Education* (pp. 433-437).
11. Capretz, L. F. (2003). Personality types in software engineering. *International Journal of Human-Computer Studies*, 58(2), 207-214.
12. Capretz, L. F., & Ahmed, F. (2010). Why do we need personality diversity in software engineering? *SIGSOFT Software Engineering Notes*, 35(2), 1-11.

13. Catanio, J. T. (2006). An interdisciplinary practical approach to teaching the software development life-cycle. *Proceedings of the 7th Information Technology Education Conference (Vol. 2006, pp. 3-8).*
14. Cherry, S., & Robillard, P. N. (2008). The social side of software engineering-A real ad hoc collaboration network. *International Journal of Human-Computer Studies, 66(7), 495-505.*
15. Creighton, O., & Singer, M. (2008). Who leads our future leaders? on the rising relevance of social competence in software development. *Proceedings of the 30th International Conference on Software Engineering (pp. 23-25).*
16. Curran, W. S. (2003). Teaching software engineering in the computer science curriculum. *ACM SIGCSE Bulletin, 35(4), 72-75.*
17. Dahiya, D. (2010). Teaching software engineering: a practical approach. *SIGSOFT Software Engineering Notes, 35(2), 1-5.*
18. Dannelly, R. S., & Garrison, C. P. (2008). Development of a graduate software project management degree. *Proceedings of the 46th Annual Southeast Regional Conference (pp. 446-449).*
19. Dawson, R. (2000). Twenty dirty tricks to train software engineers. *Proceedings of the 22nd International Conference on Software Engineering (pp. 209-218).*
20. Devlin, M., & Phillips, C. (2010). Assessing competency in undergraduate software engineering teams. *2010 IEEE Education Engineering Conference, EDUCON 2010 (pp. 271-278).*
21. Downey, J. (2005). A framework to elicit the skills needed for software development. *Proceedings of the ACM SIGMIS CPR Conference on Computer Personnel Research (pp. 122-127).*
22. Dwolatzky, B. (2004). Software engineering education in South Africa. *Transactions of the South African Institute of Electrical Engineers, 95(4), 226-235.*
23. Feldt, R., Torkar, R., Angelis, L., & Samuelsson, M. (2008). Towards individualized software engineering: empirical studies should collect psychometrics. *Proceedings of the 30th International Conference on Software Engineering. In: International Workshop on Cooperative and Human Aspects of Software Engineering (pp. 49-52)*
24. Ghezzi, C., & Mandrioli, D. (2005). The challenges of software engineering education. *Proceedings of the 27th International Conference on Software Engineering (pp. 637-638)*
25. Guinan, P. J., Coopriider, J. G., & Faraj, S. (1998). Enabling Software Development Team Performance During Requirements Definition: a Behavioral Versus Technical Approach. *Information Systems Research, 9(2), 101-125.*
26. Guo, J. (2009). Group projects in software engineering education. *Journal of Computing Sciences in Colleges, 24(4), 196-202.*
27. Hadjerrouit, S. (2005). Constructivism as guiding philosophy for software engineering education. *ACM SIGCSE Bulletin, 37(4), 45-49.*
28. Hall, T., Jagielska, D., & Baddoo, N. (2007). Motivating developer performance to improve project outcomes in a high maturity organization. *Software Quality Control, 15(4), 365-381.*
29. Hazzan, O., & Dubinsky, Y. (2007). Why software engineering programs should teach agile software development. *SIGSOFT Software Engineering Notes, 32(2), 1-3.*
30. Henderson, P. B. (2003). Mathematical reasoning in software engineering education. *Communications of the ACM, 46(9), 45-50.*

31. Herbsleb, J. D., & Mockus, A. (2003). An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on Software Engineering*, 29(6), 481-494
32. Hogan, J. M., & Thomas, R. (2005). Developing the software engineering team. *Proceedings of the 7th Australasian Conference on Computing Education* (pp. 203-210). IEEE Computer Society.
33. Jansma, P. A. (2009). Got software? What managers and engineers need to know. *IEEE Aerospace and Electronic Systems Magazine*, 24(4), 25-35
34. Jazayeri, M. (2004). The education of a software engineer. *Proceedings of the 19th International Conference on Automated Software Engineering* (p. xviii - xxvii).
35. Kaplan, R. M., & Tan, J. (2010). Real world experiences in a software engineering course. *ACM Conference on Information Technology Education* (pp. 89-94).
36. Largent, D. L., & Lüer, C. (2010). "You mean we have to work together!?!": A study of the formation and interaction of programming teams in a college course setting. *Proceedings of the International Computing Education Research Workshop* (pp. 41-49).
37. LeJeune, N. F. (2008). A real-world simulation technique for forming software development teams in a capstone course. *Journal of Computing Sciences in Colleges*, 24(1), 247-253.
38. Lethbridge, T. C. (2000). What knowledge is important to a software professional? *Computer*, 33(5), 44-50.
39. Li, Y., Chang, K.-C., Chen, H.-G., & Jiang, J. J. (2010). Software development team flexibility antecedents. *Journal of Systems and Software*, 83(10), 1726-1734.
40. Liew, C. W. (2005). Teaching software development skills early in the Curriculum through software engineering. *Proceedings of the 10th Annual Conference on Innovation and Technology in Computer Science Education* (Vol. 37, pp. 133-137).
41. Martínez, L. G., Licea, G., Rodríguez-Díaz, A., & Castro, J. R. (2010). Experiences in software engineering courses using psychometrics with RAMSET. *Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education* (pp. 244-248).
42. McGuire, E. G., & La Salle, A. J. (1997). Educating the 21st century software professional: breaking paradigms and building bridges for global quality. *Portland International Conference on Management and Technology* (pp. 569-572).
43. Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869-1890.
44. Morgan, J. N. (2005). Why the software industry needs a good ghostbuster. *Communications of the ACM*, 48(8), 129-133.
45. Orsted, M. (2000). Software development engineer in Microsoft: a subjective view of soft skills required. *Proceedings of the 22nd International Conference on Software Engineering* (pp. 539-540).
46. Ovans, R. (2005). All we really need to know about software engineering is in the film office space. *SIGSOFT Software Engineering Notes*, 30(4), 1-3.
47. O'Connor, R., & Coleman, G. (2002). Strategies for personal process improvement a comparison. *ACM Symposium on Applied Computing* (pp. 1036-1040)
48. Paulisch, F., & Zimmerer, P. (2010). A role-based qualification and certification program for software architects: An experience report from Siemens. *Proceedings of the 32nd International Conference on Software Engineering* (Vol. 2, pp. 21-27).

49. Pieterse, V., Kourie, D. G., & Sonnekus, I. P. (2006). Software engineering team diversity and performance. *Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries* (Vol. 204, pp. 180-186)
50. Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3), 303-337.
51. Purvis, M., Purvis, M., & Cranefield, S. (2004). Educational experiences from a Global Software Engineering (GSE) project. *Proceedings of The 6th Australasian Conference on Computing Education* (pp. 269-275). Darlinghurst, Australia: Australian Computer Society.
52. Rivera-Ibarra, J. G., Rodriguez-Jacobo, J., Fernández-Zepeda, J. A., & Serrano-Vargas, M. A. (2010). Competency framework for software engineers. *Proceedings of the 23rd Conference on Software Engineering Education and Training* (pp. 33-40)
53. Robillard, P. N., & Robillard, M. P. (2000). Types of collaborative work in software engineering. *Journal of Systems and Software*, 53(3), 219-224.
54. Rombach, D., Münch, J., Ocampo, A., Humphrey, W. S., & Burton, D. (2008). Teaching disciplined software development. *Journal of Systems and Software*, 81(5), 747-763.
55. Ruff, S., & Carter, M. (2009). Communication learning outcomes from software engineering professionals: A basis for teaching communication in the engineering curriculum. *Proceedings of the 39th Frontiers in Education Conference* (pp. 1271-1276).
56. Schneider, J.-G., Johnston, L., & Joyce, P. (2005). Curriculum development in educating undergraduate software engineers - Are students being prepared for the profession? *Proceedings of the 7th Australian Software Engineering Conference* (Vol. 2005, pp. 314-323).
57. Seffah, A., & Grogono, P. (2002). Learner-centered software engineering education: from resources to skills and pedagogical patterns. *Proceedings of the 15th Software Engineering Education and Training* (pp. 14-21).
58. Sharp, H., Baddoo, N., Beecham, S., Hall, T., & Robinson, H. (2009). Models of motivation in software engineering. *Information and Software Technology*, 51(1), 219-233.
59. Steen, O. (2007). Practical knowledge and its importance for software product quality. *Information and Software Technology*, 49(6), 625-636.
60. Stroustrup, B. (2010). What should we teach new software developers? why? *Communications of the ACM*, 53(1), 40-42.
61. Surakka, S. (2007). What subjects and skills are important for software developers? *Communications of the ACM*, 50(1), 73-78.
62. Teague, J. (1998). Personality type, career preference and implications for computer science recruitment and teaching. *Proceedings of the 3rd Australasian Conference on Computer Science Education* (pp. 155-163).
63. Tucker, A. (2002). On the balance between theory and practice. *IEEE Software*, 19(5), 94-96.
64. Turley, R. T., & Bieman, J. M. (1995). Competencies of exceptional and non exceptional software engineers. *Journal of Systems and Software*, 28(1), 19-38.
65. Walker, E. L., & Slotterbeck, O. A. (2002). Incorporating realistic teamwork into a small college software engineering curriculum. *Journal of Computing Sciences in Colleges*, 17(6), 115-123.

66. Wang, Z. (2009). Team, leadership, ethic, and profession in software engineering education. World Congress on Software Engineering (Vol. 4, pp. 81-83).
67. Whitehead, J. (2007). Collaboration in Software Engineering: A Roadmap. Future of Software Engineering - FOSE (pp. 214-225)
68. Yang, C. (2010). The classification and cultivating software engineers. Proceedings of International Conference on Artificial Intelligence and Education (ICAIE) (pp. 560-563).

Apêndice B

Termo de Consentimento Livre e Esclarecido

Prezado(a) Senhor(a),

O grupo de pesquisa USES eventualmente realiza estudos experimentais para caracterizar/avaliar uma determinada tecnologia de software. Estes estudos são conduzidos por alunos de Pós-graduação em Informática da Universidade Federal do Amazonas (UFAM), sob a orientação da Profa. Tayana Conte. Você foi previamente selecionado pelo seu perfil/conhecimento/experiência e está sendo convidado a participar desta pesquisa. Essa pesquisa será feita com base em dados coletados a partir de trabalhos práticos. Embora o trabalho prático faça parte da disciplina, você tem o direito de **não** permitir a utilização dos dados do seu trabalho na pesquisa.

1. Procedimentos

O estudo será realizado com data e hora marcada com os participantes pré-selecionados. Para participar do estudo normalmente será aplicado um formulário de caracterização de perfil, a fim de identificar seu nível de conhecimento/experiência. Em seguida, o estudo será executado de forma individual ou em grupos formados, seguindo sempre o planejamento do estudo feito pelo pesquisador(a) responsável. Caso seja necessário, ao final do estudo será solicitado ao participante que responda um questionário de avaliação sobre a tecnologia de software que está sendo caracterizada/avaliada.

2. Tratamento de possíveis riscos e desconfortos

Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato.

3. Benefícios e Custos

Espera-se que, como resultado deste estudo, você possa aumentar seus conhecimentos, de maneira a contribuir para o aumento da qualidade das atividades com as quais você trabalhe ou possa vir a trabalhar. Este estudo também contribuirá com resultados importantes para a pesquisa de um modo geral para o grupo de pesquisa USES. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à autorização dos seus dados na pesquisa.

4. Confidencialidade da Pesquisa

Toda informação coletada neste estudo é confidencial e seu nome não será identificado de modo algum, a não ser em caso de autorização explícita para este fim. Quando os dados forem coletados, seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados.

5. Participação

Sua participação neste estudo é muito importante e voluntária, pois requer a sua aprovação para utilização dos dados coletados neste estudo. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades. Em caso de você decidir se retirar do estudo, favor notificar o pesquisador(a) responsável. Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas. Coordenadora do Grupo de Pesquisa USES: Profa. Tayana Conte – tayana@icomp.ufam.edu.br

6. Declaração de Consentimento

Declaro que li e estou de acordo com as informações contidas neste documento e que toda linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente, recebendo respostas para todas as minhas dúvidas. Confirmo também que recebi uma cópia deste Termo (TCLE), compreendo que sou livre para não autorizar a utilização dos meus dados neste estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e concordo de espontânea vontade em participar deste estudo.

Apêndice C

Roteiro da entrevista

1. De que forma você avalia o envolvimento da turma com a atividade desenvolvida?
2. Como você avalia a satisfação dos alunos com a atividade desenvolvida?
3. Você acredita que com essa atividade, os alunos foram estimulados a desenvolver algum tipo de habilidade? Quais?
4. Quais as suas impressões gerais sobre o experimento?
5. Você conseguiu enxergar alguma vantagem na utilização de DDS em sala de aula? Quais?
6. E desvantagens, você percebeu? Quais?
7. Você acredita que houve algum fato que prejudicou o andamento da atividade? Se sim, quais?
8. Você modificaria algo na atividade realizada? O que?
9. Que outras atividades você acredita que podem ser utilizadas com DDS em sala de aula?
10. Você já havia trabalhado com DDS em sala de aula? Se sim, descreva algumas dessas experiências.
11. Como você avalia a utilização de DDS em sala de aula em comparação às abordagens tradicionais de ensino de ES?

Apêndice D

Questionário utilizado no Estudo Experimental

1. Na Tabela abaixo, circule **um número** de acordo com o quanto você concorda ou discorda de cada afirmação apresentada.

#	Afirmações	Avaliação					Comentários		
		Discordo Totalmente	-2	-1	0	+1		+2	Concordo Totalmente
1	Esta experiência contribuirá para minha vida profissional	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente	
2	A experiência ajudou no meu aprendizado	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente	
3	Esta abordagem com times distribuídos, foi mais útil ao aprendizado que as atividade co-aloçadas	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente	
4	Acredito que a atividade me propiciou uma experiência muito próxima de um cenário real de desenvolvimento	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente	
5	Acredito que a atividade me levou a perceber a importância de transmitir informações corretas para chegar ao resultado final	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente	
6	Acredito que a atividade me fez perceber a importância da colaboração para se atingir o resultado final	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente	
7	Eu gostaria de participar de novas atividades com DDS porque foi eficiente para o aprendizado	Discordo Totalmente	-2	-1	0	+1	+2	Concordo Totalmente	

2. Marque com um X quais opções abaixo representam dificuldades que sua equipe enfrentou durante a atividade. Marque quantas opções desejar.

- () Dificuldade para entender a atividade
- () Dificuldade para especificar os casos de uso
- () Dificuldade para elaborar os diagramas de casos de uso
- () Dificuldades relacionadas a tecnologia utilizada
- () Dificuldade de fazer com que o resto da equipe me entendesse
- () Dificuldade de fazer com que a equipe remota entendesse minha equipe durante as discussões sobre o documento de requisitos que foi passado a equipe

- () Quais? _____

Utilize as linhas abaixo para descrever de que forma você percebeu estas dificuldades. Você pode justificar ou descrever situações.

3. Na lista abaixo, quais itens você acredita que foram aperfeiçoados a partir da realização da atividade. Marque quantos itens julgar necessário.

- () Conhecimento sobre especificação de casos de uso e diagramas
- () Auto-disciplina
- () Habilidades de Comunicação
- () Capacidade de abstração
- () Responsabilidade
- () Habilidades de Colaboração
- () Conhecimento sobre Desenvolvimento Distribuído de Software (DDS)
- () Outros? Quais? _____

4. Que aspectos você enxerga como positivos na atividade realizada, no que diz respeito a sua aprendizagem?

Apêndice E

Lista de Acrônimos e Abreviaturas

- **DDS:** Desenvolvimento Distribuído de Software – modalidade de desenvolvimento no qual os membros das equipes encontram-se geograficamente dispersos
- **ES:** Engenharia de Software – sub-área de Ciência da Computação na qual este trabalho se concentra
- **GSE:** *Global Software Engineering* – corresponde ao DDS com um nível de dispersão global, ou seja, os membros das equipes estão em países diferentes
- **GT:** *Grounded Theory* – Teoria Fundamentada em Dados, trata-se de uma técnica de pesquisa qualitativa
- **GQM:** *Goal, Question and Metric* – abordagem utilizada em medições de software
- **MOCC:** *Motivators, Outcomes, Characteristics, and Context* – modelo proposto por Sharp et al. [2009]
- **MPS:** Melhoria do Processo de Software