

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**DESENVOLVIMENTO DE UM CONJUNTO DE
BOAS PRÁTICAS PARA A PROGRAMAÇÃO
EM PAR DISTRIBUÍDA**

BERNARDO JOSÉ DA SILVA ESTÁCIO

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Ciência da
Computação na Pontifícia Universidade
Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Rafael Prikladnicki

Porto Alegre
2013

Dados Internacionais de Catalogação na Publicação (CIP)

E79d

Estácio, Bernardo José da Silva

Desenvolvimento de um conjunto de boas práticas para a programação
em par distribuída / Bernardo José da Silva Estácio. – Porto Alegre, 2013.
107 p.

Diss. (Mestrado) – Fac. de Informática, PUCRS.

Orientador: Prof. Dr. Rafael Prikladnicki.

1. Informática. 2. Engenharia de Software. 3. Desenvolvimento
Distribuído de Software. I. Prikladnicki, Rafael. II. Título.

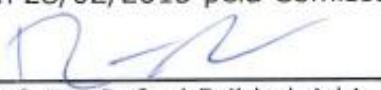
CDD 005.1

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

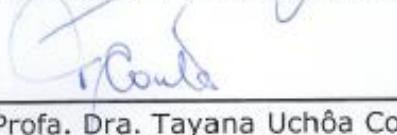
Dissertação intitulada "Desenvolvimento de um Conjunto de Boas Práticas para a Programação em Par Distribuída" apresentada por Bernardo José da Silva Estácio como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Engenharia de Software e Banco de Dados, aprovada em 28/02/2013 pela Comissão Examinadora:


Prof. Dr. Rafael Prikladnicki -
Orientador

PPGCC/PUCRS

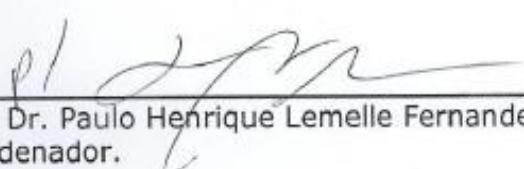

Prof. Dr. Duncan Dubugras Alcoba Ruiz -

PPGCC/PUCRS


Profa. Dra. Tayana Uchôa Conte -

UFAM

Homologada em 11/06/2013, conforme Ata No. 010, pela Comissão Coordenadora.


Prof. Dr. Paulo Henrique Lemelle Fernandes
Coordenador.

À minha avó Beatriz e para minha namorada Kamila

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus, pois até aqui tem me ajudado com Sua bondade.

O meu agradecimento especial vai para a minha namorada Kamila Baltazar, a viabilidade desta pesquisa tem o teu incentivo maior. Obrigado por ter acreditado em mim e ter compreendido os momentos em que precisei estar ausente. Sem dúvidas, essa caminhada seria mais difícil, caso tu não estivesses ao meu lado. Amo-te!

Agradeço também aos meus pais pelo apoio, imagino que tenha sido duro ver o seu filho único sair de casa tão cedo. Obrigado por compreenderem e mandarem forças, mesmo de tão longe. Não poderia deixar de citar a minha avó Beatriz, a qual sempre torceu pela minha felicidade e sucesso.

Os dois anos foram de muito aprendizado e isso em grande parte devo ao professor Dr. Rafael Prikladnicki. Obrigado por ter me selecionado e ter depositado a confiança necessária. As críticas construtivas e as sugestões foram pontuais para a concepção dessa pesquisa. Obrigado!

Quando ficamos longe de casa, os amigos se tornam a nossa família provisória, as amizades que construí em Porto Alegre foram essenciais nessa jornada. Gostaria de agradecer ao Claiton Marques, Lucas Oleksinski, Viviane Lara, Luciana Espindola, Lucas Hilgert e ao Eli Maruani. Vocês são demais!

Outro grande amigo que fiz foi um colega do grupo de pesquisa, Roni Orsoletta. Muito obrigado pela parceria, pelas sugestões e por revisar o meu texto. Não poderia deixar de citar outros colegas desta caminhada: Vanessa Gomes e Paulo Rech.

Gostaria também de agradecer ao professor Dr. Duncan Ruiz pela oportunidade que me concedeu de realizar um estágio de docência em sua disciplina. Foram momentos de muito aprendizado que ajudarão bastante a minha carreira acadêmica. Obrigado!

À HP *Enterprise Services*, por ter financiado esta pesquisa e ter me proporcionado uma estrutura adequada. Aos amigos de bolsa: Eduardo Spies, Thiago Paes, Samuel Souza. Vocês são demais! Aos ensinamentos que obtive com ótimas profissionais da indústria como a Taisa Novello e a Maria Claudia Machado. Muito obrigado!

Ao amigo Renan Sales, por ter compartilhado o apartamento comigo durante esse tempo. Obrigado por não se importar com a minha bagunça!

Aos meus amigos de Belém, agradeço por compreenderem minha ausência e sempre torcerem pelo meu sucesso. Não me esquecerei de vocês!

Às organizações participantes dos estudos de caso, por ter acreditado na contribuição desta pesquisa.

Meu muito obrigado para todos os profissionais entrevistados, pelo tempo dispensado e contribuição dada.

A todos que me ajudaram durante este período, caso não tenha mencionado o nome, deixo o meu muito obrigado.

DESENVOLVIMENTO DE UM CONJUNTO DE BOAS PRÁTICAS PARA A PROGRAMAÇÃO EM PAR DISTRIBUÍDA

RESUMO

As organizações vêm distribuindo suas atividades de desenvolvimento de software em todo o mundo há mais de uma década, aumentando o trabalho com equipes geograficamente distribuídas. Ao mesmo tempo, os métodos ágeis de desenvolvimento de software têm sido recentemente utilizados pelos engenheiros de software com o objetivo de fornecer resultados mais rápidos e de maior valor para o negócio do cliente, promovendo uma comunicação face a face, resposta rápida às mudanças, entre outras práticas. Apesar de soar contraditório, os métodos ágeis têm sido utilizados como uma estratégia para tornar equipes distribuídas mais produtivas. A programação em par é uma prática ágil do método *extreme programming*, e que tem sido utilizada com equipes distribuídas. Esta prática possui diversos benefícios, entre eles o compartilhamento de informações e o aumento da qualidade do produto. Por esta razão, o objetivo desta dissertação de mestrado é entender as vantagens e os desafios da programação em par distribuída e desenvolver um conjunto de boas práticas para facilitar a sua adoção e utilização. Para o desenvolvimento desta pesquisa foram utilizados estudos secundários (revisão sistemática da literatura) e primários (múltiplos estudos de caso com profissionais da indústria). Esta pesquisa contribui no sentido de propor um conjunto de boas práticas para a programação em par distribuída para a indústria, além da sistematização da base empírica do estado da arte sobre o tema.

Palavras chaves: Programação em par, desenvolvimento distribuído de software, *extreme programming*, programação em par distribuída.

DEVELOPMENT OF A SET OF BEST PRACTICES FOR DISTRIBUTED PAIR PROGRAMMING

ABSTRACT

Organizations have been distributing their software development activities around the world for over a decade, increasing the work with distributed teams. At the same time, agile methods have recently been used by software engineers in order to deliver faster results and more value to the client, providing face to face communication, rapid response to change, among other practices. Although it sounds contradictory, agile methods have been used as a strategy for distributed teams become more productive. Pair programming is an agile practice of the extreme programming method, which has been used with distributed teams. In this context this practice has many benefits, including information sharing and increasing product quality. For this reason, the goal of this dissertation is to understand the advantages and challenges of distributed pair programming and to develop a set of best practices to facilitate their adoption and use. For the development of this research we have used both secondary (systematic literature review) and primary (multiple case studies with practitioners) studies. The main contribution of this research is the development of a set of best practices for distributed pair programming for the industry, and the systematization of the empirical evidence about this topic.

Keywords: **Pair programming, distributed software development, extreme programming, distributed pair programming.**

LISTA DE FIGURAS

FIGURA 2.1 – CICLO DE DEMING [BAS08]	18
FIGURA 2.2 – ETAPAS DA REVISÃO SISTEMÁTICA	29
FIGURA 2.3 – TELA PRINCIPAL DO CITEULIKE.....	31
FIGURA 2.4 – ABORDAGEM DE PESQUISA DOS ARTIGOS	34
FIGURA 2.5 – MÉTODOS PESQUISAS UTILIZADOS	35
FIGURA 2.6 – TIPO DOS ARTIGOS	36
FIGURA 2.7 – ANO DE PUBLICAÇÃO DOS ARTIGOS IDENTIFICADOS	36
FIGURA 2.8 – CONTEXTO DAS PUBLICAÇÕES.....	37
FIGURA 2.9 – TIPO VEÍCULO ONDE OS ARTIGOS FORAM PUBLICADOS	38
FIGURA 2.10 – LISTA DE VARIÁVEIS EM TORNO DA PRÁTICA DE PP	41
FIGURA 2.11 – LISTA DE VARIÁVEIS EM TORNO DE PP NO ENSINO.....	44
FIGURA 2.12 – LISTA DE VARIÁVEIS EM TORNO DA PRÁTICA DE PPD.....	46
FIGURA 2.13 – LISTA DE VARIÁVEIS EM TORNO DE PPD NO ENSINO	48
FIGURA 3.1 – DESENHO DE PESQUISA	56
FIGURA 6.1 – CICLO DE MAFRA [MAF06]	84

LISTA DE TABELAS

TABELA 2.1 – <i>CHECKLIST</i> DE AVALIAÇÃO DE QUALIDADE DO ARTIGOS	32
TABELA 2.3 – LISTA DE VARIÁVEIS EM TORNO DA PRÁTICA DE PP	40
TABELA 2.4 – LISTA DE VARIÁVEIS EM TORNO DE PP NO ENSINO	43
TABELA 2.5 – LISTA DE VARIÁVEIS EM TORNO DA PRÁTICA DE PPD	45
TABELA 2.6 – LISTA DE VARIÁVEIS EM TORNO DE PPD NO ENSINO	47
TABELA 4.1 – RESUMO DOS PROJETOS ANALISADOS.....	61
TABELA 4.2 – CONFIGURAÇÃO DOS PROJETOS ANALISADOS.....	68
TABELA 4.3 – CARACTERÍSTICAS DOS PARES ENTRE OS PROJETOS	69
TABELA 4.4 – BENEFÍCIOS DA PPD	71
TABELA 4.5 – DESAFIOS DA PPD.....	72
TABELA 4.6 – PONTOS OBSERVADOS NO ESTUDO.....	75
TABELA 5.1 – CONJUNTO DE BOAS PRÁTICAS	78

LISTA DE ABREVIATURAS

DDD – *Domain-Driven Design DDD*

DDS – Desenvolvimento Distribuído de Software

ES – Engenharia de Software

XP – *eXtreme Programming*

PP – Programação em Par

PPD – Programação em Par Distribuída

PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul

RSL – Revisão Sistemática da Literatura

TI – Tecnologia da Informação

TDD – *Test- Driven Development*

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVOS	14
1.2 JUSTIFICATIVA	14
1.3 ORGANIZAÇÃO DO VOLUME	15
2 REFERENCIAL TEÓRICO.....	16
2.1 DESENVOLVIMENTO DE SOFTWARE	16
2.2 DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE (DDS)	19
2.3 MÉTODOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE.....	21
2.4 PROGRAMAÇÃO EM PAR	25
2.5 PP E PPD: UMA REVISÃO SISTEMÁTICA DA LITERATURA.....	27
3 METODOLOGIA DE PESQUISA.....	55
3.1 DESENHO E ETAPAS DA PESQUISA.....	55
3.2 ASPECTOS METODOLÓGICOS	57
3.3 BASE METODOLÓGICA DO ESTUDO DE CASO	58
4 RESULTADOS DO ESTUDO DE CASO	61
4.1 CARACTERIZAÇÃO DOS PROJETOS ANALISADOS	61
4.2 CARACTERIZAÇÃO DOS RESPONDENTES E SUA PARTICIPAÇÃO.....	64
4.3 RESULTADOS DO ESTUDO DE CASO	65
4.4 LIÇÕES APRENDIDAS	73
4.5 LIMITAÇÕES DO ESTUDO DE CASO	75
5 CONJUNTO DE BOAS PRÁTICAS PARA PDD	77
5.1 CONJUNTO DE BOAS PRÁTICAS PARA PPD	77
5.2 LIMITAÇÕES DO CONJUNTO DE BOAS PRÁTICAS PROPOSTO	82
6 CONCLUSÃO	83
6.1 CONTRIBUIÇÕES DA PESQUISA	83
6.2 TRABALHOS FUTUROS	83
REFERÊNCIAS BIBLIOGRÁFICAS	85
APÊNDICE A.....	91
APÊNDICE B.....	96
APÊNDICE C	102

1 INTRODUÇÃO

A crescente globalização vivenciada nas últimas décadas tem causado impacto em diferentes ramos da indústria, e consequentemente, no desenvolvimento de software. No cenário de busca de mercados mais vantajosos, menor custo de mão de obra, melhor qualidade do produto a ser entregue ao cliente, entre outros fatores, as empresas da área de TI passaram a distribuir seus processos de desenvolvimento de software [SMI12]. Foi neste contexto que no final da década de 90 surgiu o Desenvolvimento Distribuído de Software (DDS) que é caracterizado quando um projeto de software é desenvolvido por equipes que possuem colaboradores fisicamente distantes entre si [HER01].

Quase na mesma época, em 2001, surgiu o Manifesto Ágil e os métodos ágeis vieram à tona [BEC01], tendo como principal característica uma abordagem adaptativa voltada para ambientes com requisitos voláteis. Por seu caráter inovador, os métodos ágeis têm atraído interesses tanto da academia quanto da indústria até os dias de hoje [Dyb08]. *Extreme Programming* (XP) é considerado um dos métodos ágeis mais utilizados na indústria [BEC04] [Dyb08], possuindo uma série de técnicas e práticas que apóiam atividades de desenvolvimento de software, ao contrário de outros métodos ágeis, como, por exemplo, o Scrum que tem seu enfoque mais voltado para práticas de gerência de projetos [SCH95].

A Programação em Par (PP) é uma prática ágil do método XP e consiste na cooperação entre dois desenvolvedores utilizando um mesmo computador para desenvolver software [MCD02]. Alguns estudos anteriores mostraram que a PP é uma área com pontos positivos e pontos negativos a serem levados em consideração [Dyb09]. Entre os pontos positivos é possível citar: a qualidade do código, o auxílio no aprendizado e a produtividade. Nos pontos negativos: a personalidade não colaborativa de alguns programadores, o tipo de tarefa ser conduzida para prática e a infraestrutura necessária.

Por conta da adoção de práticas ágeis com equipes distribuídas, emergiu-se o conceito da Programação em Par Distribuída (PPD) [PAA09] [BAH02]. Na PPD dois desenvolvedores colaboram juntos no desenvolvimento de software por meio de um ambiente ferramental que possibilite o compartilhamento de tela e um meio de comunicação como áudio, texto e vídeo.

A PPD vai de encontro a alguns princípios estabelecidos em PP como a comunicação face a face e a colaboração no mesmo ambiente físico. Neste contexto, os benefícios oriundos da PPD passam a ser questionados e vários desafios em torno da prática são identificados. Sendo assim, esta pesquisa tem por objetivo propor um conjunto

de boas práticas para PPD que facilite a adoção e utilização da prática, além de identificar os seus benefícios e desafios.

Desta forma, a questão de pesquisa que norteou este estudo foi:

Quais são as boas práticas recomendadas para viabilizar a programação em par distribuída e quais seus benefícios e desafios?

1.1 Objetivos

O **objetivo geral** dessa pesquisa foi propor um conjunto de boas práticas para a utilização da Programação em Par Distribuída. De forma a complementar o objetivo geral proposto, os seguintes **objetivos específicos** foram definidos:

- Aprofundar os estudos da base teórica em métodos ágeis e desenvolvimento distribuído de software;
- Definir e executar uma revisão sistemática da literatura com os estudos sobre a prática de programação em par e programação em par distribuída;
- Elaborar um conjunto de boas práticas preliminar da Programação em Par Distribuída;
- Planejar e realizar múltiplos estudos de caso;
- Propor um conjunto de boas práticas para a Programação em Par Distribuída.

1.2 Justificativa

Nos anos recentes os métodos ágeis têm atraído interesses tanto da academia quanto da indústria, com o método XP sendo considerado um dos mais utilizados [Dyb08] [Ver11]. A programação em par mostrou ser uma efetiva prática sendo utilizada de maneira pedagógica nas universidades [Mcd02] e com muitos benefícios na indústria [Wil00].

Uma recente *survey*, publicada em 2011 pela Version One Inc.[Ver11] mostrou que 65% dos 6.042 participantes (profissionais da indústria de software) pertencem a um time de desenvolvimento distribuído de software (DDS) [Ver11]. Esta *survey* é realizada anualmente pela Version One Inc desde 2006, e nos dois últimos anos (2010 e 2011) o percentual de participantes pertencentes a uma equipe de DDS aumentou. Neste contexto crescente de DDS, há um crescente interesse da indústria na utilização de práticas ágeis, entre elas a Programação em Par Distribuída (PPD). Este interesse está associado ao fato de que as práticas ágeis podem ajudar equipes de DDS a minimizar parte dos desafios que elas possuem ao desenvolver software de forma distribuída [Paa09].

Alguns estudos analisaram a prática de PPD, como consta Canfora *et al.* [CAN03] [CAN06] e Rosen *et al.* [ROS10]. Apesar destes estudos analisarem os efeitos da PP em ambientes distribuídos e também apontarem lições aprendidas sobre a prática. A literatura específica não possui estudos que reúnam boas práticas que facilitem a execução desta atividade em um contexto de DDS.

Desta forma, esta pesquisa se justifica por uma necessidade da academia e da indústria. Na academia, resultados empíricos são necessários para a avaliação da prática de PPD, no contexto de entendimento dos efeitos empíricos da área de Métodos Ágeis e DDS. Já a indústria se situa em um contexto crescente de DDS, onde são necessários subsídios, apoio para adoção, utilização e compreensão dos efeitos de práticas de desenvolvimento ágil de software para DDS, como é o caso da PPD.

1.3 Organização do Volume

Este volume está organizado em 6 capítulos. Na sequência, o Capítulo 2 apresenta o referencial teórico desta pesquisa, envolvendo os principais conceitos e implicações das áreas de estudos: desenvolvimento de software, desenvolvimento distribuído de software, *extreme programming*. Neste capítulo também é apresentada a revisão sistemática da literatura (RSL) de PP e PPD executada, apresenta-se o protocolo que orientou a revisão e os principais resultados obtidos.

No Capítulo 3, apresenta-se a metodologia de pesquisa, descrevendo cada uma das etapas do estudo, justificando a escolha e explicando o uso dos métodos utilizados.

No Capítulo 4, descrevem-se os múltiplos estudos de caso, desenvolvidos com 14 profissionais com experiência em PPD. A pesquisa, alicerçada nos resultados da revisão sistemática, aborda os principais desafios, benefícios e estratégias de sucesso dos profissionais que implementaram PPD.

O conjunto de boas práticas para PPD proposto é apresentado no Capítulo 5, como consequência do processo de pesquisa seguido, apoiado nos resultados da revisão sistemática da literatura e da pesquisa de campo realizada.

Finalmente, o Capítulo 6 apresenta as considerações finais desta dissertação, enfatizando as contribuições e limitações. Conclui-se destacando as oportunidades futuras de pesquisas a partir deste trabalho.

2 REFERENCIAL TEÓRICO

O referencial teórico constitui uma importante etapa da pesquisa, contendo os principais conceitos e teorias das áreas pesquisadas. Na seção 2.1 apresenta-se o conceito e a evolução do desenvolvimento de software. Após, na seção 2.2 será abordado o desenvolvimento distribuído de software. A seção 2.3 apresenta os métodos ágeis para desenvolvimento de software, caracterizando o método ágil *Extreme Programming* que está diretamente relacionado ao tema desta pesquisa. Na seção 2.4 apresenta-se a Programação em Par e a Programação em Par Distribuída. Finalmente, a seção 2.5 apresenta os resultados de uma revisão sistemática da literatura realizado sob as duas práticas.

2.1 Desenvolvimento de software

Em ambientes de mercado heterogêneos e nos mais variados domínios de conhecimento, o software tem se tornado imprescindível. O desenvolvimento de software como produto tem como base a Engenharia de Software (ES), que segundo a IEEE [PRE11] é conceituada como sendo a “aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software”. Ela constitui uma disciplina da engenharia que engloba todos os aspectos da produção de software [SOM11].

Segundo Pressman [PRE11], a Engenharia de Software pode ser representada em camadas. A base destas camadas é o foco na qualidade, ratificando o compromisso na qualidade do desenvolvimento de software. A camada dos métodos fornece a técnica de “como” fazer o software, já a camada de ferramentas provê um apoio automatizado ou semi-automatizado para os métodos.

O alicerce é a camada de processo, pois é a base para o controle gerencial de projetos de software e estabelece o contexto no qual os métodos e as ferramentas serão aplicados. Esta camada faz a ligação com as outras camadas e permite o desenvolvimento oportuno e racional do software [PRE11].

Um processo de software reúne uma série de atividades e resultados associados à produção de software [SOM11]. Pode também ser referenciado como ciclo de vida do software, pois descreve a “vida” do produto de software desde a concepção até a implementação, entrega e a manutenção [PFL04].

A representatividade e a descrição de um processo são feita por meio de modelos. Cada modelo possui características únicas, podendo estas serem semelhante na teoria,

mas diferentes na prática [PFL04]. Os modelos são constituídos das atividades do processo, dos papéis, métodos da ES e ferramentas utilizadas no apoio.

De acordo com Sommerville [SOM11], existem atividades que são comuns a todos os modelos de processo de software, quais sejam:

- Especificação de Software: clientes e a equipe definem o produto a ser desenvolvido e as restrições existentes;
- Desenvolvimento de Software: projeção e desenvolvimento do software;
- Validação de Software: onde é feita a validação do que foi produzido com a conformidade do que foi solicitado;
- Evolução do Software: adaptação às mudanças dos requisitos do cliente e de mercado.

A escolha de um determinado modelo de processo para desenvolvimento de software deve ser realizada mediante a algumas condições que devem ser consideradas, tais como: a natureza do projeto e da aplicação, os métodos e as ferramentas a serem utilizados, os controles e os produtos a serem entregues [PRE11]. Embora não exista um modelo “ideal”, cabe ressaltar que existe a possibilidade dos modelos serem aprimorados no contexto de diferentes organizações [SOM11].

2.1.1 Modelos prescritivos de processo de software

Os modelos prescritivos de processo foram os primeiros modelos propostos na literatura da Engenharia de Software, também denominados como modelos convencionais (ou ainda tradicionais), que tinham por finalidade colocar ordem no caos do desenvolvimento de software [PRE11].

Segundo Pressman [PRE11] esses modelos são ditos “prescritivos” porque possuem em sua estrutura uma ordem formal de elementos do processo, tais como atividades, ações da ES, tarefas, produtos de trabalho, mecanismos de garantia de qualidade, controle de modificações para cada projeto. Além disso, possuem um fluxo de trabalho que descreve como cada um destes elementos se relaciona uns com os outros. A principal característica desses modelos é a busca pela estrutura e ordem [PRE11].

Os modelos prescritivos possuem atividades genéricas e a diferença se faz na invocação de cada atividade e nas respectivas ações de ES implementadas para cada modelo. Apesar da literatura também apresentar o termo tradicional para esses modelos, não significa dizer que não são mais utilizados no mercado [SOM11].

2.1.2 Modelos adaptativos de processo de software

Segundo Keen um modelo adaptativo consiste em três componentes: o usuário, o analista e o próprio sistema, os quais durante o processo de construção do produto interagem mutuamente [KEE80]. A ideia da adaptabilidade é enraizada nos princípios da melhoria contínua provenientes, sobretudo, das indústrias automotivas.

Um dos exemplos disso é o ciclo PDCA (*Plan-Do-Check-Act*) ou ciclo de Deming, o qual foi proposto nos anos 50 baseado nos estudos de modelo estatístico de controle de qualidade de Shewhart na década de 20. O ciclo (figura 2.1) é proposto por quatro etapas: Planejamento, Execução, Verificação e Ação. A fase de melhoria contínua no processo está na Ação, onde ocorre a investigação dos problemas, a correção deles e a tomada de ações para que estes não aconteçam novamente por meio da melhora do processo de trabalho [BAS08].



Figura 2.1 – Ciclo de Deming [BAS08]

Modelos adaptativos não são processos rígidos, mas são focados no produto, nas pessoas que o desenvolvem e são abertos a mudanças (“*embrace the change*”) [PRE11]. No contexto de software, esses modelos podem apoiar o desenvolvimento, principalmente em ambientes onde os requisitos do produto são voláteis. Não tão determinístico como os modelos prescritivos de processo, os adaptativos possuem forte embasamento no empirismo, isto é, na experiências vivenciadas na prática [BAS08] [PRE11]. Os métodos ágeis são exemplos conhecidos de modelo adaptativo de processo.

Independente de fazer uso de modelos prescritivos ou adaptativos, as empresas têm buscado novas formas para desenvolver software, em busca de vantagens competitivas como a diminuição de custos e o aumento da capacidade de produção.

Dessa forma, o desenvolvimento de software pode ocorrer com equipes distribuídas em outros locais ou países. A próxima seção apresenta o desenvolvimento distribuído de software (DDS), que está ganhando cada vez mais destaque neste contexto.

2.2 Desenvolvimento distribuído de software (DDS)

A globalização dos negócios, amparada pela sofisticação dos meios de comunicação, possibilitou que muitas organizações pudessem distribuir seus processos dentro dos seus próprios países, ou para outros países, ou regiões visando, sobretudo, à redução de custos [AUD07]. Este cenário, não é diferente para o desenvolvimento de software e tem impulsionado consideráveis investimentos no Desenvolvimento Distribuído de Software (DDS).

O DDS é caracterizado sempre que um ou mais recursos humanos envolvidos no projeto estiver fisicamente distantes dos demais [AUD07]. Carmel [CAR99] conceitua o DDS como um modelo de desenvolvimento onde os envolvidos em um determinado projeto estão dispersos, sendo este diferenciando do co-localizado quanto à dispersão geográfica, temporal e diferenças culturais. Pode-se dizer também que uma equipe global de desenvolvimento de software está tipicamente em países diferentes, porém colaborando em um mesmo projeto de qualquer natureza (criação ou manutenção de software) [LAN08].

2.2.1 Fatores que levam ao DDS

Os principais fatores que contribuíram para o crescimento do DDS nos últimos anos [CAR99], [PRI03], [FRE05], [SMI12] foram:

- A necessidade de recursos globais para serem utilizados a qualquer hora;
- Incentivos fiscais para o investimento em pesquisas de informática;
- Disponibilidade de mão de obra especializada e de custos reduzidos em países em desenvolvimento;
- Vantagem de estar perto do mercado local, incluindo o conhecimento dos clientes e as condições locais para explorar as oportunidades de mercado;
- Rápida formação de organizações e equipes virtuais para explorar as oportunidades locais;
- Grande pressão para o desenvolvimento *time-to-market* (velocidade no trabalho, tempo entre a concepção e a comercialização do produto) utilizando as vantagens do fuso horário diferente, no desenvolvimento conhecido como

follow-the-sun (24 horas contínuas, cotando com equipes fisicamente distribuídas);

- Necessidade de integrar recursos resultantes de aquisições e fusões organizacionais.

2.2.2 Níveis de dispersão

As equipes no DDS podem estar em diferentes níveis de dispersão quanto as distâncias geográficas. Audy e Prikladnicki [AUD07] indicam quatro situações que verificam o tipo de distância física e suas características principais:

- Mesma localização física: a empresa possui todos os atores em um mesmo local, as reuniões ocorrem sem dificuldades em relação à distância e a equipe pode interagir estando fisicamente presente. Não existe diferença de fuso horário e as diferenças culturais raramente envolvem a dimensão nacional, sendo que os obstáculos são mesmos vistos no desenvolvimento centralizado de software;
- Distância nacional: nesse caso os atores estão localizadas dentro de um mesmo país, podendo reunir-se em curtos intervalos de tempo. Dependendo do país, pode haver diferenças em relação ao fuso horário e cultura;
- Distância continental: essa situação as equipes estão localizadas em países diferentes, mas dentro do mesmo continente. As reuniões face a face entre as equipes ficam mais difíceis de serem realizadas devido a distância física. O fuso horário também pode exercer um papel importante na equipe, podendo dificultar algumas interações;
- Distância global: nessa situação os atores estão localizadas em países diferentes e em continentes diferentes, formando muitas vezes uma distribuição global. Os encontros face a face geralmente ocorrem no início dos projetos. Fatores como, por exemplo, a diferença cultural, comunicação e fuso horário exercem um papel fundamental, podendo impedir interações entre as equipes.

Em relação à dimensão relacionada com a distância geográfica, a distribuição ocorre através de duas formas principais, sendo *onshore*, que ocorre no mesmo país onde estão localizados o cliente e a matriz da empresa, e *offshore*, quando a matriz da empresa contratante está em outro país em relação ao cliente.

2.2.3 Desafios do DDS

Apesar dos fatores que impulsionam a adoção do DDS, é importante também notar os desafios e problemas causados. Com a distribuição do processo de desenvolvimento de software em diversos locais novos problemas surgem, como por exemplo, como garantir uma comunicação efetiva em profissionais espalhados ao redor do mundo ou como coordenar equipes em diferentes fusos horários.

Segundo Audy [AUD07], os desafios de DDS se dividem em cinco categorias: Pessoas (diferenças culturais, idiomas); Processo (a forma como o processo será definido no DDS, podendo ser um desenvolvimento adaptativo ou processos prescritivos); Tecnologia (recursos tecnológicos devem existir para facilitar a comunicação e colaboração da equipe); Gestão (está ligada à atividades de gerência de projetos como gerenciamento de risco, alocação de recursos, entre outros) e a Comunicação (ter uma comunicação efetiva mesmo com dispersão geográfica e temporal).

2.3 Métodos ágeis de desenvolvimento de software

Quase na mesma época do surgimento do DDS, os métodos ágeis de desenvolvimento de software se tornaram populares a partir do Manifesto Ágil¹ publicado em 2001. Estes métodos hoje têm sido utilizados também em um contexto de DDS.

Conforme mencionado na Seção 2.1, os métodos ágeis possuem as características dos modelos adaptativos [BAS08]. O Manifesto Ágil reúne os princípios dos métodos ágeis:

1. **Indivíduos e iterações** são mais importantes que processos e ferramentas.
2. **Software funcionando** é mais importante do que documentação completa.
3. **Colaboração com o cliente** é mais importante do que negociação de contratos.
4. **Adaptação a mudanças** é mais importante do que seguir o plano inicial.

Os itens valorizados à esquerda do manifesto enfatizam a característica adaptativa que permeiam os métodos ágeis, tais como: a adaptação a mudanças, o foco no produto e a comunicação entre os colaboradores e com o cliente.

Os diversos métodos ágeis existentes possuem princípios em comuns [SOM11], também baseados no manifesto:

¹ <http://agilemanifesto.org>

- Envolvimento do cliente: os clientes devem estar bem envolvidos com o processo de desenvolvimento. O seu papel é priorizar e prover novos requisitos do sistema e avaliar as iterações;
- Satisfazer o cliente: o software deve possuir valor e ser entregue de forma adiantada;
- Interação entre Pessoas de Negócio e o Time: devem trabalhar em conjunto e diariamente durante todo o projeto;
- Entrega incremental: esta característica não é exclusiva dos métodos ágeis, consiste no desenvolvimento de software por meio de incrementos, onde o cliente especifica quais requisitos devem ser incluídos em cada incremento;
- Software funcional: representa a medida primária de progresso;
- Conversa cara a cara: representa o método mais eficiente e eficaz de transmitir informações dentro de um time de desenvolvimento;
- Excelência técnica e bom design: contínua atenção para excelência técnica e bom design aumentam a agilidade;
- Time motivado: os membros do time desenvolvimento devem possuir um ambiente e o suporte necessário para realizar suas atividades. As habilidades de desenvolvimento do time devem ser desenvolvidas e exploradas;
- Abraçar as mudanças: espera os requisitos do sistema para mudanças e então projetar o sistema de acordo com as mudanças;
- Manter a simplicidade: focar na simplicidade do software desenvolvido e no seu processo de desenvolvimento, efetivamente trabalhar para eliminar a complexidade do sistema.
- Times auto-organizáveis: as melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis;
- Intervalos Regulares de reflexão: o time reflete em como ficar mais efetivo e aprimora e otimiza o comportamento.

Nesse sentido, alguns métodos ágeis presentes na indústria como: XP (*eXtreme Programming*), Crystal Clear [COC04], Scrum [SCH95] e *Lean Software Development* [POP03], ressaltam os princípios da agilidade no processo de software, tendo como característica a adaptabilidade as mudanças nos requisitos dos projetos de desenvolvimento de software. A seguir serão apresentadas as características do método XP, que é o método ágil que compõe o contexto desta pesquisa.

2.3.1 Extreme programming

Extreme Programming (XP) é um dos métodos ágeis mais conhecidos e utilizados na indústria [SOM11]. Ele foi proposto por Kent Beck por meio de sua experiência na indústria de software [BEC04]. O nome XP foi dado por Beck em vista à abordagem desenvolver as boas práticas já conhecidas, como o desenvolvimento iterativo, para níveis “extremos”. Por exemplo, na Programação Extrema, várias versões de um sistema podem ser desenvolvidas por diferentes programadores, integradas e testadas em até um dia [SOM11].

O projeto da Chrysler foi o pioneiro na formação do método, nele Beck reuniu práticas como: revisão de códigos, teste, integração rápida, *feedback* do cliente, design simples, entre outras que proveram uma melhoria na qualidade do produto. A sua proposta inovava em utilizar estas práticas de forma extrema por meio de aprimoramentos, como por exemplo: revisão de códigos com programação em pares, testes utilizando testes automatizados e antecipando sua execução. Isto trouxe grande repercussão, pois divergia do cenário de desenvolvimento de software da época [BAS08].

O método XP possui em seu alicerce valores, os quais o permeiam, sendo estes:

- Comunicação: é fundamental no alinhamento do produto com os requisitos do cliente. Os princípios da humanidade e da diversidade se preocupam com a comunicação entre os desenvolvedores [BAS08];
- Simplicidade: evitar funcionalidades e complexidades desnecessárias para codificação. Possui como principais princípios: melhorias, auto-semelhança e passos pequenos [BAS08];
- Coragem: este valor está relacionado com a inovação e com as mudanças, no decorrer de um projeto às vezes é necessário tomar decisões para ser conseguir as metas e os objetivos proposto [BAS08];
- *Feedback*: quanto mais rápido os impedimentos forem identificados e corrigidos, por menos tempos estes causaram empecilhos ao projeto. Aliada a comunicação dos envolvidos o *feedback* possibilita que a equipe e o projeto identifiquem os problemas e possam se adaptar a eles [BAS08];
- Respeito: todos os membros da equipe devem manter o respeito entre si com o propósito de manter a sinergia de grupo. No XP é considerado o lado humano dos desenvolvedores, onde os princípios da pessoalidade e da diversidade são alinhados com a aceitação da responsabilidade [BAS08].

Os valores do XP apontam para uma característica comum entre a maioria dos métodos ágeis, os aspectos humanos. Cockburn [COC06] corrobora a esta idéia ao dizer que “o desenvolvimento ágil enfoca os talentos e as habilidades dos indivíduos moldando o processo a pessoas e equipes específicas” [PRE11].

Práticas

O método XP possui 13 práticas primárias [BEC04]:

1. Desenvolvimento dirigido por testes: Testes unitários são escritos para todos os componentes do sistema e são automatizados pelos desenvolvedores antes da implementação da funcionalidade. Testes de regressão, integração e aceitação também desempenham um papel importante no método XP.
2. Trabalho Energizado: O trabalho não deve ser exaustivo ao ponto de afetar a vida pessoal e a saúde dos participantes. Uma sobrecarga de trabalho diminui a qualidade do trabalho, portanto, esta prática recomenda que as horas de trabalho sejam calculadas de uma forma realística.
3. Integração Contínua: O código fonte é mantido em um repositório comum a toda equipe, assim sempre que alguma tarefa for concluída o novo código é integrado ao repositório. Com a gerência do repositório, todos os desenvolvedores se mantém sincronizados, trabalhando na última versão do sistema.
4. Sentar-se Junto: A equipe XP deve trabalhar em lugar amplo e aberto, onde todos possam ficar juntos. O propósito desta maior proximidade entre as pessoas é contribuir para a comunicação, colaboração e ajuda mútua.
5. Time Completo: A equipe XP deve ser composta não só por desenvolvedores, mas por analistas de testes, analistas de negócios, clientes, especialistas em banco de dados, etc. Desta a equipe XP deve possuir uma natureza multidisciplinar.
6. Área de Trabalho Informativa: No ambiente de trabalho deve se ter informações sobre o andamento do projeto, dados do projeto como métricas de andamento devem estar visíveis, isto permitirá que a equipe identifique questões relevantes e tomem decisões com base nisso.
7. Histórias: As funcionalidades são brevemente escritas pelo cliente em cartões de papel, os quais podem ser manuseados por eles e pelos desenvolvedores.
8. Ciclo Semanal: Esta prática sugere que a equipe planeja o trabalho de cada iteração durante uma semana. Por semana, os membros se reúnem para:

refletir sobre o progresso realizado, planejar e priorizar as histórias planejadas pelo cliente, realizando assim, a fase de planejamento.

9. Ciclo Trimestral: Por trimestre as *releases* são produzidas, o nível de planejamento é de versões entregáveis do software por meio várias iterações.
10. Folga: No plano é incluso algumas tarefas menores que podem ser removidas, caso ocorra um atraso. A folga permite que os desenvolvedores fiquem mais confortáveis para assumir os compromissos e o cliente tenha segurança sobre os prazos de entrega estabelecidos.
11. *Build* Ágil: Uma *build* consiste na execução automatizada dos testes para verificação de código com o propósito de colocar o software em funcionamento. Quanto mais rápido for a *build*, o código será executado mais vezes e o *feedback* para a equipe será mais freqüentem, aumentando assim a descoberta de erros.
12. *Design* incremental: O *design* do sistema deve prezar pela simplicidade, tendo como suporte a refatoração e os testes automatizados de forma a garantir a resolução de problemas futuros com rapidez.
13. Programação em par: O desenvolvimento do sistema é realizado por duas pessoas trabalhando no mesmo computador, com um teclado e um mouse [BEC04]. As duplas são trocadas com frequência, tanto em nível de papel quanto aos membros de cada par.

2.4 Programação em par

A programação em par é uma das práticas primárias do método XP. Como o nome sugere, é uma prática de programação que envolve dois programadores em um mesmo computador trabalhando de forma colaborativa [MCD02]. Um programador se comporta como *driver* e implementa o código controlando o teclado e o mouse. Outro programador se comporta como *observer* ou *navigator* e é responsável por revisar o código ao mesmo tempo, prevenir e identificar erros lógicos e sintáticos no código. Ambos os programadores podem atuar nos dois papéis e decidir a frequência de troca [MCD02].

Além da colaboração, a comunicação é um dos requisitos da programação em par. Ambos os programadores devem estar em constante contato para discutir possíveis soluções e erros de código [HAO11]. O sucesso da aplicação desta prática possui relação direta com o local de trabalho, este deve ser aberto e facilitar a comunicação entre os pares [VAN07b]. A viabilidade do uso da prática de programação em par está relacionada

também com a complexidade da tarefa [DYB09]. Programação em par não melhora a produtividade em tarefas rotineiras, triviais ou de teste [DYB09].

Os benefícios da prática de programação em par reportado são vários, entre eles: aumento da produtividade, aumento da qualidade de produto devido ao alto numero de defeitos encontrados na revisão contínua de um dos pares, o aumento da colaboração e comunicação do time e o aprimoramento da condição de trabalho (confiança e motivação) que não deixa que um programador se sinta isolado [HAO11].

Muitos estudos envolvendo a prática de programação em par já foram realizados tanto no contexto educacional quanto na indústria. No contexto educacional, os resultados mostraram que a prática cria um ambiente que conduz de forma mais avançada os estudantes a um aprendizado ativo e a interação social, aumentando mais as notas, confiança e o interesse na área de computação [SAL11]. Na indústria, os resultados relatam o aumento da qualidade do código e a desempenho no trabalho, entretanto algumas críticas também foram relatadas como o aumento de gasto com esforço, custo de pessoal e os conflitos de personalidade entre os programadores [DYB09]. Mais vantagens e desvantagens sobre PP serão detalhados na seção seguinte.

2.4.1 Programação em par distribuída

A adoção de DDS pelas organizações implica na investigação de como as práticas de desenvolvimento de software são utilizadas com equipes distribuídas. Com a estratégia de adoção de Métodos Ágeis no DDS, a PP se torna alvo desse contexto, sendo uma prática ágil que exige comunicação face a face e uma colaboração com a interação constante entre os programadores. A distribuição geográfica levanta alguns desafios nesses aspectos.

Baheti atestou que os benefícios de PP são os mesmos em PPD, tais como a produtividade e qualidade do código [BAH02]. Outro benefício de PPD é que dada as suas características, ela ajuda à promoção do trabalho e da comunicação dentro de equipes distribuídas [BAH02]. Contudo, o foco aos aspectos de infraestrutura deve ser redobrado, e isto se reflete na adoção de um ferramental específico para a prática. A principal condição para que PPD funcione é o uso de uma ferramenta específica. Ho aponta que o uso de ferramentas específicas para PPD ajuda a tornar a prática mais rápida [HO04]. Canfora corrobora dizendo que uma ferramenta específica evita que o programador fique alternando entre diferentes ferramentas, gastando tempo [CAN06]. Este ferramental tem que prover um bom canal de comunicação por meio de chat e

divisão da área de trabalho entre os programadores. Canfora [CAN06] em seu estudo relatou que a qualidade do código sofre uma pequena diminuição devido as questões de infraestrutura.

Outro ponto a ser abordado é a idéia de time, com a distância geográfica são necessárias estratégias para que todos conheçam o projeto de maneira igual para que não haja empecilhos no desenvolvimento do sistema e não afetar a colaboração entre os pares. A colaboração também pode enfrentar outros obstáculos como o fuso horário, a cultura e o idioma [CAN06]. Quanto ao esforço não há evidências que relatam um aumento [CAN06].

A PPD também mostrou ser uma prática que gera benefícios no ensino de programação. Alguns efeitos positivos foram encontrados em relação ao Aprendizado, Qualidade do Código [ZAC11] e ao Desempenho do Programador [HAN05]. Em um experimento realizado por Hanks [HAN05], os alunos que realizaram a PPD tiveram um desempenho tão bom quanto os alunos que estavam com pares co-locados, passando no curso com notas similares. A próxima seção terá um detalhamento maior sobre as variáveis e efeitos em torno dos estudos da prática e do ensino em PPD.

2.5 PP e PPD: uma revisão sistemática da literatura

Apesar de alguns estudos como Canfora [CAN03, CAN04] e Rosen [ROS10] analisarem PPD e levantarem algumas lições aprendidas sobre a prática, poucos são os estudos que exploram e propõe boas práticas para PPD. Uma revisão inicial da literatura da área, realizada no início dessa pesquisa, em 2011, identificou somente experimentos e propostas de ferramentas para PPD. Desta forma, após a fase exploratória de pesquisa, optou-se pela utilização de um método secundário de pesquisa, a Revisão Sistemática da Literatura (RSL) visando melhor identificar e caracterizar propostas de boas práticas para PPD.

A RSL é parte da Engenharia de Software baseada em evidências proposta por Kitchenham [KIT07]. É um importante processo de identificação, avaliação e análise de evidências disponíveis com o propósito de responder questões de pesquisas específicas. A revisão sistemática seguiu as orientações de Kitchenham, onde dois pesquisadores participaram do processo de revisão[KIT07]. Um aluno de mestrado foi o principal revisor, o qual foi responsável por desenvolver o protocolo de pesquisa, estratégia de busca, a seleção dos estudos, a avaliação da qualidade dos estudos, a extração e síntese dos dados e os relatórios dos resultados encontrados. O outro revisor foi responsável por validar o protocolo de revisão, monitorar os resultados da busca e o processo de seleção

e verificar os dados da avaliação de qualidade e as informações de extração dos dados. A seguir apresentam-se o planejamento e resultados obtidos nesta atividade.

2.5.1 Protocolo da RSL

A revisão sistemática da literatura foi executada seguindo as recomendações documentadas por Kitchenham [KIT07] e outras experiências de RSL na literatura com tema relacionado [DYB08, SAL11]. Desta forma, a condução da RSL seguiu etapas definidas estrategicamente (Figura 2.2), cada etapa será explicada, em detalhes, nas próximas seções.

Um protocolo de revisão sistemática (apêndice A) que possuía as diretrizes para a condução da RSL foi criado. O objetivo da revisão era responder as seguintes questões de pesquisa:

- QP1: O que se sabe sobre a utilização da programação em par e a utilização da programação em par distribuída?
- QP2: Em que condições a programação em par funciona?
- QP3: Em que condições a programação em par distribuída funciona?

2.5.2 Identificação da literatura relevante

A primeira tarefa na identificação da literatura presente na área é a construção de uma *string* de busca. Desta forma, as estratégias para construção dos termos da pesquisa utilizadas foram:

- Derivação de termos utilizados na questão de pesquisa (exemplo: estratégias, benefícios, limitações);
- Listar as palavras-chave dos primeiros artigos consultados de PP;
- Uso do operador booleano OR para incorporar sinônimos;
- Uso do operador AND para fazer a junção entre as diferentes palavras-chave.

A *string* de busca inicialmente utilizada para a pesquisa foi:

(“Agile Software Project” OR “Agile Software Development” OR “eXtreme Programming” OR “XP”) AND (“Pair programming” OR “programming practice” OR “pair-programming”) AND (“Limitations” OR “Best Practices” OR “Benefits” OR “Advantages” OR “Disadvantages” OR “Design” OR “Strategies”).

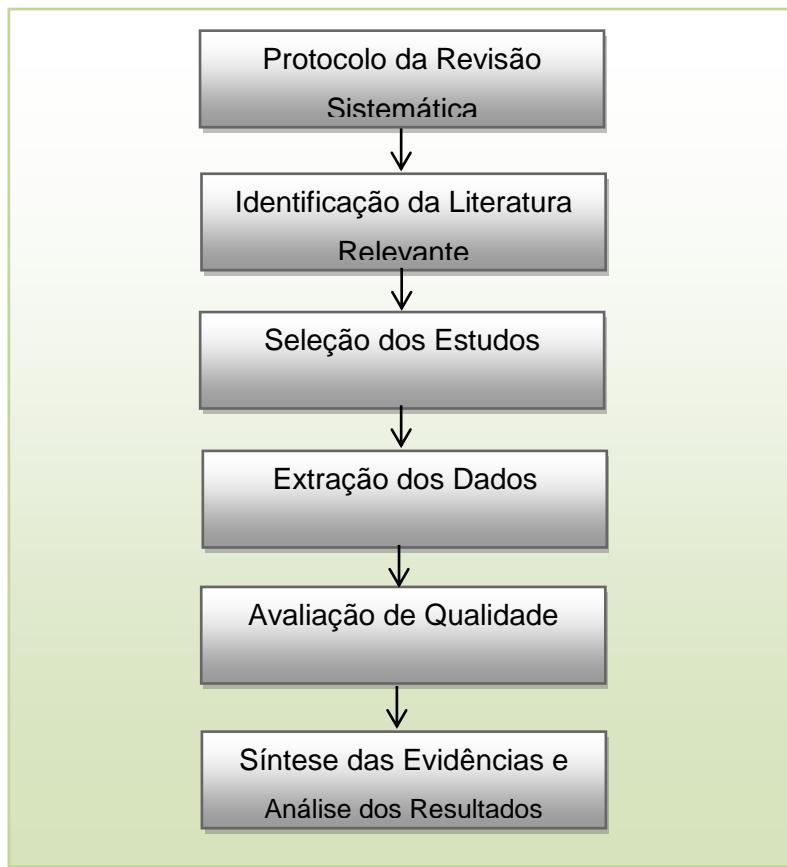


Figura 2.2 – Etapas da Revisão Sistemática

Ao aplicar a *string* completa em uma pesquisa preliminar, obtivemos poucos artigos. Por exemplo, a biblioteca IEEEExplore retornou apenas 15 artigos e a biblioteca Compedex retornou 36. Desta forma, optou-se por utilizar uma estratégia de uma *string* de busca mais aberta, com mais ruído, porém que retornasse mais resultados, já que na primeira tentativa importantes artigos de controle ficaram de fora. A *string* escolhida foi “*pair programming*” OR “*pair-programming*” semelhante à adotada no estudo de Dyba [DYB08] e Salleh [SAL11]. Esta *string* possibilitou além de abranger todos os estudos retornados na *string* anterior, expandir a quantidade de resultados, incluindo os artigos de controle definidos.

As bases utilizadas para a pesquisa foram:

- ACM Digital Library
- IEEE Xplore
- Compedex
- Scopus
- ISI Web of Knowledge
- Willey

A escolha por estas bases de dados *on line* se deu pelas bases utilizadas em outras revisões sistemáticas da literatura [DYB08] [SAL11]. Além disso, a escolha das bases de dados também foi feita, a partir do conhecimento dos pesquisadores sobre bases de dados que indexavam artigos de PP e PPD e por meio das bases disponibilizadas pela PUCRS.

Semelhante ao estudo de Salleh [SAL11] a base de dados referência da pesquisa selecionada foi a Scopus devido a sua reputação e grande quantidade de resumos e citações. Portanto, cada artigo retornado das outras bases foi comparado com a lista já existente da Scopus com o intuito de evitar duplicações.

2.5.3 Seleção dos estudos

O critério de inclusão tinha por objetivo apenas estudos que usassem a PP no contexto do desenvolvimento de software. A pesquisa apenas cobriu estudos que foram publicados entre 2001 e junho de 2012. A data de 2001 foi escolhida em função da publicação do manifesto ágil, a qual foi uma base para a popularização de métodos ágeis como o XP, a data fim de junho de 2012 está relacionada ao período de término de execução da revisão.

Para a exclusão de um artigo da pesquisa, os seguintes critérios foram adotados:

- Artigos publicados em eventos não acadêmicos da área de computação;
- Artigos que não abordam a prática de Programação em Par;
- Estudos que não sejam no idioma inglês;
- Tutoriais, *Short papers* e palestras.

O processo de seleção de estudos consistiu em duas fases. Na primeira fase, para cada artigo pesquisado, foi analisado o título, o resumo e as palavras chaves. Os artigos relevantes foram armazenados para uma avaliação posterior. Na segunda fase cada estudo relevante foi lido na íntegra para determinar se era um estudo primário da revisão sistemática.

2.5.4 Extração dos dados

Para a extração dos dados foi utilizado uma ficha de leitura desenvolvida no MS Excel¹. Os itens da ficha foram selecionados conforme alinhamento as questões de pesquisa. A ficha possuía os seguintes itens:

¹ <http://office.microsoft.com/pt-br/excel/>

- Artigo
- Ano
- Autor
- Veículo (onde foi publicado)
- Tipo (*Journal, Conference, Workshop*)
- Objetivo, Contexto (Educacional, Indústria)
- Contribuição, Evidências
- Ferramentas/Infraestrutura e Processo
- Trabalhos Futuros
- Metodologia Empregada
- PPD (Sim, Não)
- Status (Incluso ou Não Incluso)
- Justificativa (referente ao status)

A ferramenta web Citeulike¹ foi utilizada para gerenciar a leitura e a geração das referências dos artigos conforme mostra a figura 2.3 a seguir.

The screenshot shows the CiteULike web interface. At the top, there are tabs for 'CiteULike' and 'MyCiteULike'. On the right, it says 'Logged in as bernardoestacio' and has 'Log Out' and 'Tags' buttons. Below the tabs, the title 'My library 146 articles' is displayed with a link icon. Underneath are buttons for 'Search', 'Unwatch', 'Copy', 'Export', 'Sort', and 'Hide Details'. The main area lists 146 articles. The first few are:

- Impact of pair programming on thoroughness and fault detection effectiveness of unit test suites**
Software, Vol. 13, No. 3. (May 2008), pp. 281-295, doi:10.1002/spip.v13:3
by Lech Madeyski
posted to pp by bernardoestacio keyed Madeyski2008Impact on 2012-06-14 19:18:56 ★★★★★/
■ Abstract ■ Copy
- Towards Understanding Communication Structure in Pair Programming** [Quick Edit] [CITO]
In Agile Processes in Software Engineering and Extreme Programming, 11th International Conference, XP 2010, Trondheim, Norway, June 1-4, 2010. Proceedings, Vol. 48 (2010), pp. 117-131
by Kai Stapel, Eric Knauss, Kurt Schneider, Matthias Becker
edited by Alberto Sillitti, Angela Martin, Xiaofeng Wang, Elizabeth Whitworth
posted to pp by bernardoestacio keyed StapelKS10 on 2012-06-04 19:19:28 ★★★★★/
■ Copy
- Do programmer pairs make different mistakes than solo programmers?**
J. Syst. Softw., Vol. 80, No. 9. (September 2007), pp. 1460-1471, doi:10.1016/j.jss.2006.10.032
by Matthias M. Müller
posted to pp by bernardoestacio keyed Muller2007Do on 2012-04-25 13:18:29 ★★★★★/
■ Abstract ■ Copy
- Investigating the effective implementation of pair programming: an empirical investigation**
In Proceedings of the 42nd ACM technical symposium on Computer science education (2011), pp. 655-660, doi:10.1145/1953163.1953346
by Alex D. Radermacher, Gursimran S. Walia
posted to pp by bernardoestacio keyed Radermacher2011Investigating on 2012-04-25 12:50:32 ★★★★★/
■ Abstract ■ Copy
- Assessing distributed pair programming**
In Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (2002), pp. 50-51, doi:10.1145/985072.985099
by Prashant Baheti
posted to pp by bernardoestacio keyed Baheti2002Assessing on 2012-04-24 15:35:26 ★★★★★/
■ Abstract ■ Copy

Figura 2.3 – Tela Principal do CiteULike

¹ <http://www.citeulike.org/>

2.5.5 Avaliação da qualidade dos estudos

Para apoiar a extração de dados um *checklist* de qualidade foi desenvolvido. As perguntas e a pontuação foram adaptadas da revisão sistemática de Salleh [SAL11]. O *checklist* é composto por sete perguntas com a seguinte escala: Sim = 1 ponto, Parcialmente = 0,5 ponto e Não = 0 pontos. O resultado total de cada estudo tem uma faixa de 0 (Muito ruim) até 7 (Muito bom). A Tabela 2.1 apresenta o *checklist* de qualidade de avaliação dos artigos acompanhado das diretrizes usadas.

Tabela 2.1 – *Checklist* de Avaliação de Qualidade do Artigos

Item	Resposta
1 -O trabalho é bem/adequadamente referenciado (apresenta trabalhos relacionados/semelhantes e baseia-se em modelos e teorias da literatura)?	() Sim () Parcialmente () Não
2- O objetivo da pesquisa é claro (*)?	() Sim () Parcialmente () Não
3- O método de pesquisa foi apropriado para alcançar os objetivos da pesquisa?	() Sim () Parcialmente () Não
4- Existe uma clara descrição do contexto no qual a pesquisa foi realizada?	() Sim () Parcialmente () Não
5- A coleta de dados foi realizada adequadamente(**)?	() Sim () Parcialmente () Não
6- A análise de dados foi realizada adequadamente (***)?	() Sim () Parcialmente () Não
7-Os resultados possuem credibilidade (****)?	() Sim () Parcialmente () Não
Indicadores/Diretrizes de Qualidade	
(*) O artigo é baseado em uma pesquisa ou é apenas um conjunto um relatório de lições aprendidas baseado na opinião de um expert?	
(**) Há uma Discussão de: Quem conduziu o de coleta de dados? Procedimentos / documentos utilizados para a coleta / Áudio ou gravação de vídeo de entrevistas / debates / conversas (Se não foram registrados há uma justificativa dada?) Como os métodos de estudo de campo ou estudo de caso aplicado podem ter influenciado nos dados coletados.	
(***) Existe uma descrição profunda na análise de dados? Tem dados suficientes para apoiar os resultados? Os dados contraditórios foram levados em consideração? Métodos de controle de qualidade foram usados para verificar os resultados?	
(****) Resultados são suportados por dados / estudos empíricos (ou seja, o leitor pode ver como o pesquisador chegou a seus / suas conclusões, a metodologia da análise e interpretação são evidentes) Os resultados e considerações possuem uma lógica coerente? Uso de evidências para apoiar ou refinar resultados?	

2.5.6 Execução

A partir do protocolo de pesquisa definido, a RSL foi executada. A busca inicial com a string “*pair programming*” OR “*pair-programming*” retornou 391 artigos. Destes, apenas 147 foram considerados potencialmente relevantes baseado na leitura do título e do resumo. Todos os 147 artigos foram lidos por completo e após aplicação dos critérios de inclusão e exclusão e a remoção de duplicações foram obtidos 96 artigos para a avaliação de qualidade.

A tabela 2.2 apresenta a pontuação de qualidade dos 96 artigos. A maior parte dos artigos foi avaliada como Bom, com 25 estudos (aproximadamente 25%) e Muito Bom, com 56 estudos (aproximadamente 58%). Um artigo foi avaliado como Ruim, pois não possui evidência empírica e nem metodologia descrita. Desta forma, este artigo foi retirado da fase de análise. Portanto, ao final, **95 artigos** foram incluídos para a análise das evidências. Tabela 2.2 – Pontuação de Qualidade

Qualidade	Muito Ruim (<2)	Ruim (2 - <3)	Regular (3-<5)	Bom (5 – <=6)	Muito Bom (>6)	Total
Número de Estudos	0	1	14	25	56	96
Porcentagem	0	~11%	~15%	~26%	~58%	100%

2.5.6.1 Análise dos Resultados

Os resultados da RSL foram analisados tanto em nível quantitativo quanto qualitativo. A lista com todos os artigos analisados se encontra no apêndice B desta dissertação, para referenciá-los utilizou-se um padrão número específico com o objetivo de diferenciar das outras referências da dissertação.

2.5.6.1.1 Análise Quantitativa

A análise quantitativa foi dividida em classificação dos artigos, análise por ano, contexto das publicações, tipo de artigo, principais autores e análise da referência.

Classificação dos Artigos

A classificação usada por Wieringa [WIE06] foi utilizada como base para definir o tipo pesquisa dos 95 artigos analisados. Esta classificação está assim definida:

1. *Evaluation Research*: Técnicas ou soluções são implementadas e avaliadas na prática, e as consequências investigadas.

2. *Validation Research*: Técnicas que foram propostas, mas ainda não foram executadas na prática, é um típico estudo de laboratório.
3. *Solution Proposal*: A solução para um problema é proposta e os seus benefícios são discutidos. A diferença entre um artigo *Solution Proposal* para *Validation Research* é o tipo de abstração das soluções sugeridas, a qual possui um nível maior nos artigos *Solution Proposal*.
4. *Philosophical Paper*: Estrutura a área em forma de taxonomia ou framework conceitual.
5. *Experience Paper*: Inclui a experiência pessoal do autor na percepção de como ocorreu na prática.
6. *Opinion Paper*: A opinião pessoal do autor sobre um problema sem trabalhos relacionados e métodos de pesquisa.

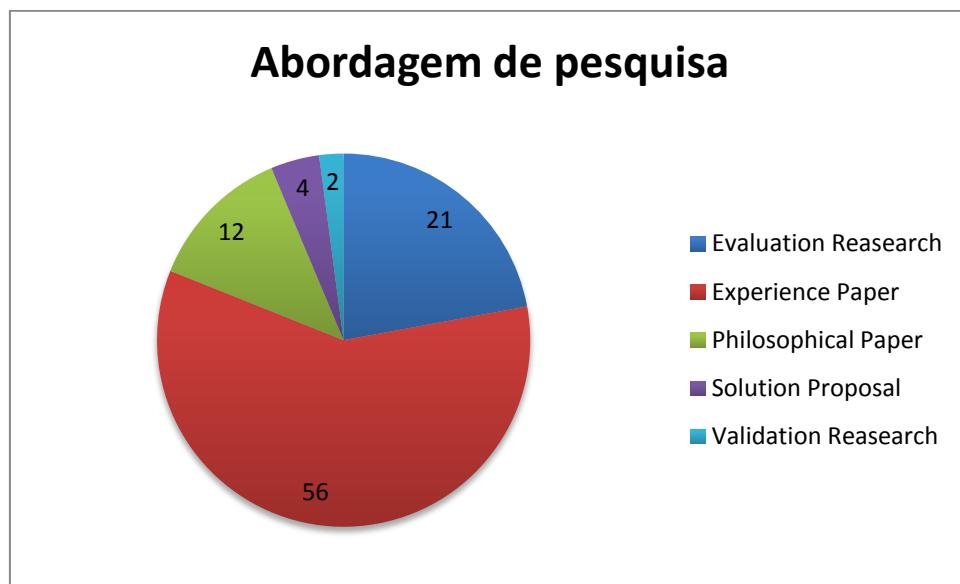


Figura 2.4 – Abordagem de pesquisa dos artigos

Baseado na análise (figura 2.4), a abordagem mais encontrada foi a de *Experience Paper* (53 estudos). Artigos da abordagem *Evaluation Research* que descrevem estudos de casos é a segunda classificação mais presente com 20 estudos. **Método de Pesquisa**

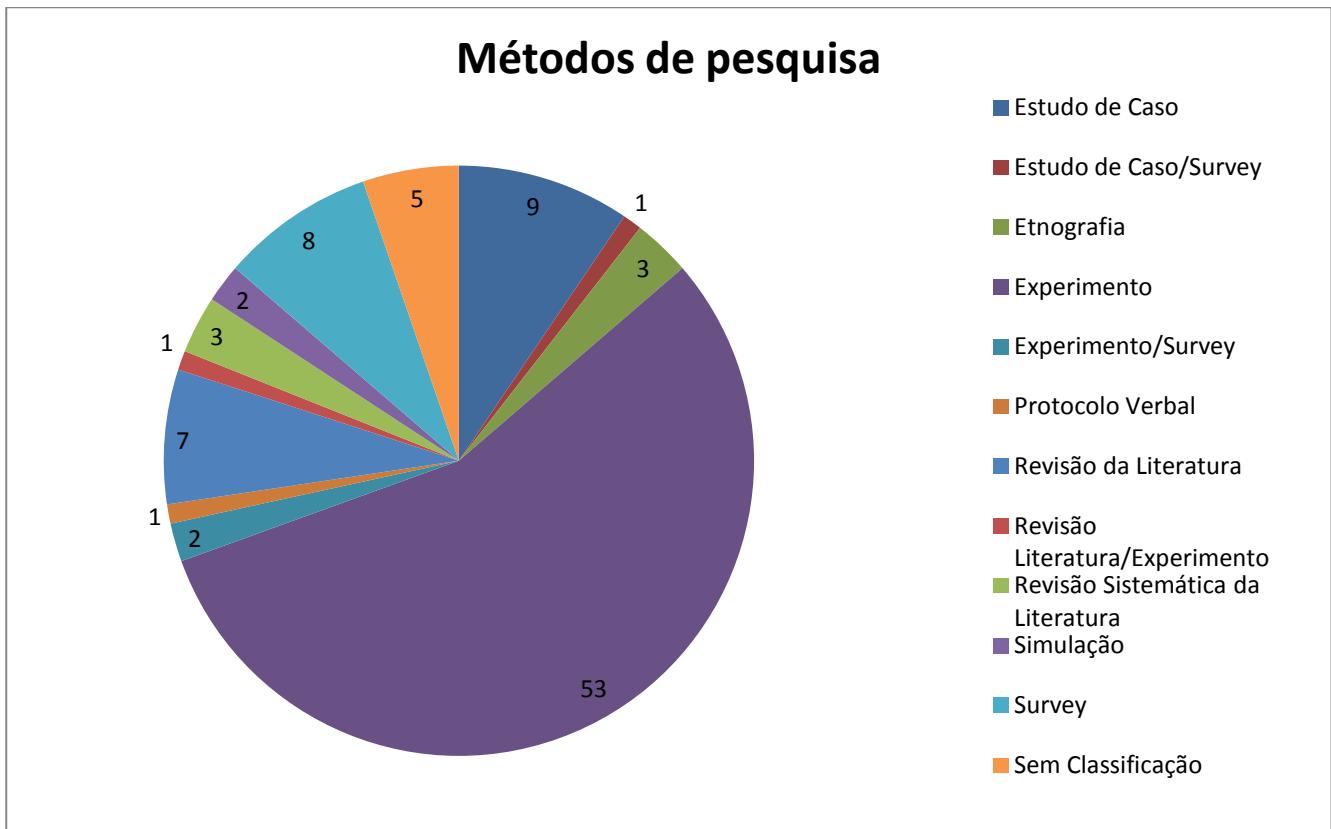


Figura 2.5 – Métodos pesquisas utilizados

Em 53 estudos o método de pesquisa utilizado foi o experimento (figura 2.5). Alguns métodos foram usados em conjunto com outros, tais como Estudo de Caso e Survey e Experimento e Survey.

Tópico Abordado

As questões de pesquisa da revisão sistemática abrangem tanto a PP quanto PPD. A figura 2.6 apresenta o tipo dos artigos analisados, sendo que a maior parte dos artigos é referente a PP, com 73 estudos.



Figura 2.6 – Tipo dos Artigos

Análise por Ano

A distribuição dos artigos ao longo dos anos é apresentada na figura 2.7. No ano de 2012 os artigos foram analisados até o mês de junho (mês que a revisão foi executada). O primeiro artigo que trata sobre PPD é de 2002.

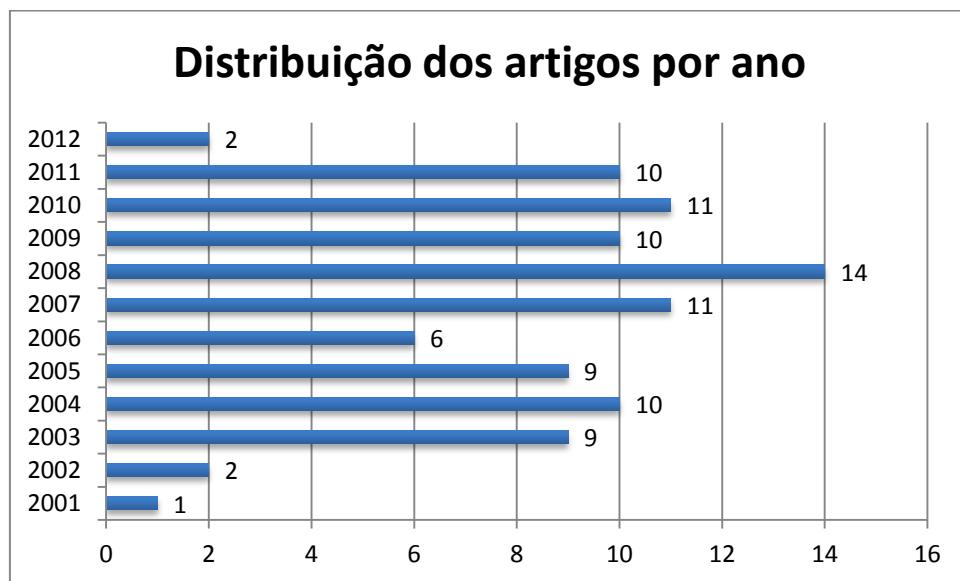


Figura 2.7 – Ano de publicação dos artigos identificados

Contexto das Publicações

As publicações foram classificadas em três categorias (figura 2.8):

- Ensino: Os estudos que abordam os efeitos da PP e PPD como ferramenta pedagógica para ensino de programação.

- Prática: Os estudos que abordam os efeitos da PP e PPD como prática no desenvolvimento de software.
- Ferramentas, Modelos, Frameworks: Os estudos que descrevem modelos, *frameworks* e ferramentas de suporte para PP e PPD. Alguns destes estudos possuem avaliações sobre o que fora proposto, por exemplo, estudos com ferramentas têm experimentos utilizando a ferramenta proposta.

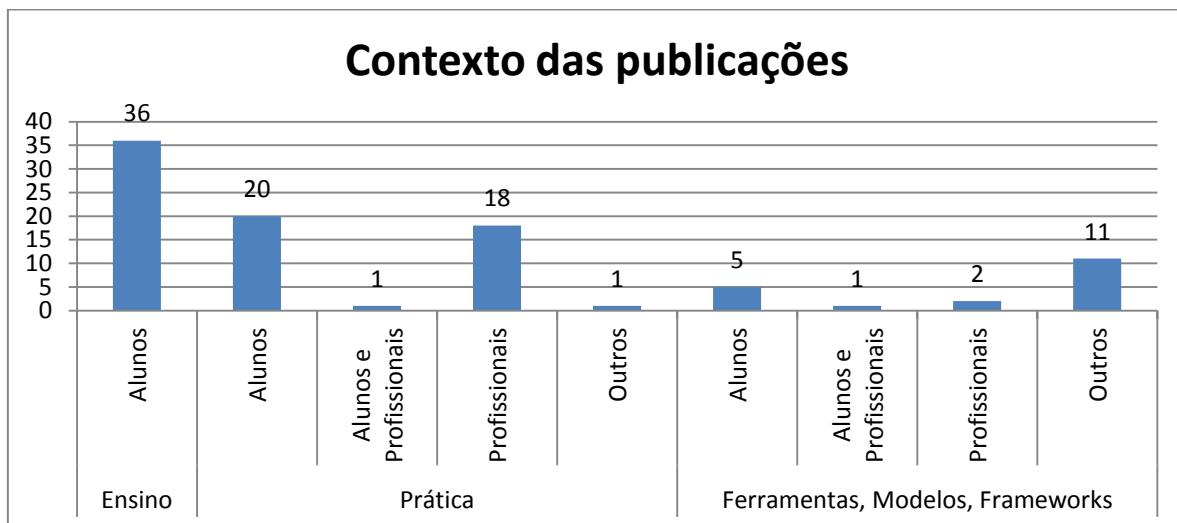


Figura 2.8 – Contexto das Publicações

Dentro de cada categoria, foi classificado ainda o tipo da amostra (Alunos, Profissionais, Outros) que o estudo utilizou. Dos noventa e quatros de estudos analisados, 39 buscaram analisar PP e PPD, 36 estudos buscaram analisar efeitos de PP e PPD no ensino e 19 analisaram ou propuseram ferramentas, modelos e *frameworks* para PP e PPD. Esta revisão sistemática enfocou mais nos estudos com evidências empíricas: Ensino e Prática de PP e PPD.

Análise por tipo veículo de publicação

Quanto ao tipo de meio de publicação dos estudos, a predominância são os artigos publicados em conferências, sendo 68 estudos deste tipo. Apenas 21 estudos foram publicados em *Journal* e 7 em *Workshops*. A figura 2.9 mostra o tipo de veículo onde os artigos foram apresentados.

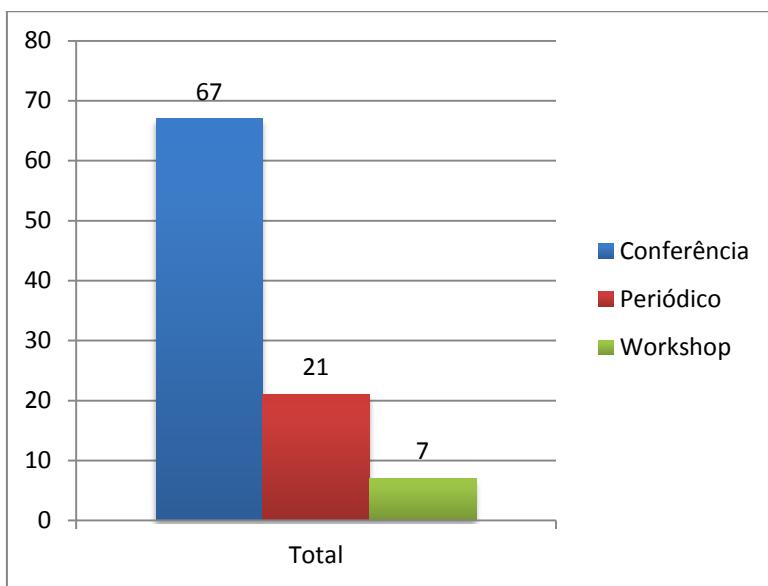


Figura 2.9 – Tipo veículo onde os artigos foram publicados

Especificamente para PPD apenas 3 estudos foram publicados em Journal, enquanto 4 estudos foram publicados em Workshop, sendo este número quase o total da amostra. Estes números apresentam indícios de poucos resultados científicos publicados sobre PPD e a presença de estudos em estágios mais iniciais.

2.5.6.1.2 Análise qualitativa

A análise qualitativa teve por objetivo caracterizar os estudos selecionados, com o propósito de identificar o conteúdo de cada artigo em vista a responder as questões de pesquisa definidas no protocolo da revisão sistemática.

O que se sabe sobre a utilização da programação em par?

A Revisão Sistemática apresentou um extenso panorama da área de PP conseguindo responder a primeira questão de pesquisa (QP1) presente no protocolo apêndice A. Do total dos estudos selecionados, 38 estudos apresentaram evidências empíricas na utilização da prática de PP, seja como um fator positivo, negativo ou sem algum efeito. Os resultados foram classificados pelo contexto atribuído na revisão.

Prática

Vinte e um estudos analisaram os efeitos que a prática de PP tem sobre o desenvolvimento de software. A tabela 2.3 abaixo mostra os estudos, o tipo de amostra utilizada e as variáveis analisadas.

A variável mais relatada pelos estudos selecionados foi a qualidade do código. Seis estudos relatam que a qualidade do código é uma variável que possui um efeito positivo na utilização de PP e apenas um estudo relata que PP não traz nenhum efeito

significativo sob essa variável. Três estudos que envolviam experimentos com alunos usaram como métrica a densidade de defeitos encontrados nos programas desenvolvidos, sendo o valor encontrado menor com equipes que utilizaram PP [MUL07], [SIS09], [VAN05]. Na indústria, dois estudos foram baseados em dados de uma *survey* aplicados em organizações [BEG08], [VAN07a]. No primeiro estudo cerca de 65,4% dos 487 respondentes disseram que PP produz código com mais qualidade [BEG08]. No outro estudo os respondentes indicaram um baixo índice de defeitos ao usar PP [VAN07a]. Ainda envolvendo dados de profissionais, Hulkko mensurou a qualidade do código com três métricas: densidade de desvios no padrão do código, densidade de defeitos e a taxa de comentários [HUL05]. Este estudo mostrou que PP não possui um efeito significativo da qualidade em comparação com a programação individual.

A complexidade da tarefa é uma variável que influencia na efetividade de PP. Quatro estudos atestam isso [ARI07], [DYB07], [DYB09], [SIS09]. Dyba através de uma meta-análise relata que PP é mais rápido que a programação individual quando a complexidade da tarefa é baixa e provê maior qualidade para tarefas de complexidade alta, porém com um esforço maior [DYB09]. Envolveando experimento com alunos, Sison corrobora ao concluir que PP é tão efetiva em projetos complexo, sem diminuir a produtividade [SIS09]. Na indústria, Giri disse que em um cenário onde se possui profissionais com pouca experiência é bom que tarefas complexas PP sejam realizadas em pares [GIR12]. Em um experimento com alunos, Vanhanen não encontrou nenhum efeito significativo nas atividades com a mudança de complexidade das tarefas [VAN05]. A produtividade foi outra variável a ser medida pelo autor, sendo a quantidade de trabalho dividida pelo esforço gasto [VAN05]. Segundo o experimento realizado por ele, PP teve 29% menos produtividade que a programação individual [VAN05]. Ele atribui isso ao tempo de aprendizado de um parceiro [VAN05].

Tabela 2.3 – Lista de variáveis em torno da prática de PP

Prática					
Tipo de Amostra	Variáveis	Efeito Positivo	Efeito Negativo	Sem Efeito	Total de Artigos
Alunos	Qualidade do Código	[SIS09], [MUL07], [VAN05]			3
	Complexidade do sistema e da tarefa	[SIS09]		[VAN05]	2
	Diferentes tipos de personalidades	[CHO07a]		[VAN10]	2
	Diferença conhecimento	[CAO05]		[MUL04]	2
	Colaboração entre as equipes	[BIP07], [VAN05]			2
	Produtividade		[VAB07]		1
	Motivação	[MUL07]			1
	Resolução de problemas pouco familiares	[LUI06]			1
	Aprendizado	[VAN05]			1
	Introdução de novas pessoas ao projeto	[LUI06]			1
Profissionais	Qualidade do Código	[BEG08], [VAN07a]		[HUL05]	3
	Complexidade do sistema e da tarefa	[ARI07], [GIR12], [WAL09]			3
	Diferentes tipos de personalidades	[WAL09]	[BEG08]		2
	Diferença conhecimento	[CHO07b]	[BEG08]		2
	Produtividade			[HUL05]	1
	Desempenho do programador	[VAN07]			1
	Tempo maior na execução da tarefa		[FRO11]		1
Outros (Revisões Sistemáticas, Revisões da Literatura)	Complexidade do sistema e da tarefa	[DYB07], [DYB09]			2
	Diferença conhecimento	[DYB07]			1

A figura 2.10 apresenta a síntese dos resultados da tabela 2.3, ilustrando o tipo e a quantidade de estudos em torno dos efeitos das variáveis da prática de PP.

Variáveis em torno da prática de PP									
Qualidade de código		[HUL05]	[SIS09]	[MUL07]	[VAN05]	[BEG08]	[VAN07]		
Complexidade do sistema e tarefa		[VAN05]	[SIS09]	[ARI07]	[GIR12]	[WAL09]	[Dyb07]	[Dyb09]	
Diferentes tipos de personalidade	[BEG08]	[HAN10]	[CHO07a]	[WAL09]					
Diferença conhecimento	[BEG08]	[MUL04]	[CAO05]	[CHO07b]	[Dyb07]				
Produtividade	[VAN07a]	[HUL05]							
Colaboração entre equipe			[BIP07]	[VAN05]					
Motivação			[MUL07]						
Resolução de problemas pouco familiares			[LUI06]						
Aprendizado			[VAN05]						
Introdução de novas pessoas ao projeto			[LUI06]						
Desempenho do programador			[VAN07b]						
Tempo maior de execução da tarefa	[FRO11]								
Legenda: Alunos	-1	0	1	2	3	4	5	6	
Profissionais		Negativo	Sem efeito	Positivo					
Outros (RSL)									

Figura 2.10 – Lista de variáveis em torno da prática de PP

A personalidade dos programadores é outra variável que impacta a efetividade de PP. Segundo Walle, baseado nas evidências de um estudo com empresas, a diferença de traços na personalidade pode aumentar a quantidade de comunicação entre um par [WAL09]. Outro estudo na indústria relata que a diferença de personalidade é um dos principais empecilhos de PP [BEG08]. Choi por meio de um experimento no ambiente educacional utilizou a classificação tipológica Myers-Briggs (MBTI) [CHO07a]. Os resultados apontaram que pares formados com tipos diferentes apresentaram mais produtividade. Hannay mostrou que a personalidade não tem um efeito significativo na PP e enfatizam que mais estudos precisam ser realizados [HAN10].

Cinco estudos relatam que o conhecimento do programador é uma importante variável em PP [BEG08], [CHO07b], [CAO05], [Dyb07], [MUL04]. Chong, por meio de

uma etnografia na indústria, observou que a diferença de conhecimento entre os pares é uma variável que tem uma forte influência na interação em PP [CHO07b]. Quando esta diferença é alta, o programador com menor conhecimento tem dificuldades de avaliar os argumentos técnicos apontados pelo programador mais experiente [CHO07b]. Em uma survey aplicada por Begel a maioria dos programadores respondentes apontou que preferem formar pares com pessoas de conhecimento semelhante [BEG08]. De acordo com Cao, diferentes combinações de pares quanto ao conhecimento podem alcançar diferentes resultados [CAO05]. Por exemplo, uma combinação com um programador experiente e um com pouca experiência ajuda no compartilhamento do conhecimento. Já uma combinação com um programador experiente formando com outro experiente ajuda na melhoria da qualidade do código e para geração de novos conhecimentos. Muller não encontrou efeito significativo do conhecimento dos programadores em relação à efetividade de PP em um experimento educacional [MUL04].

Na indústria PP mostrou ainda ter um bom desempenho na resolução de problemas pouco familiares e na introdução de pessoas novas ao projeto [LUI06]. Contudo, o tempo se mostrou como sendo um efeito negativo em PP, sendo este maior em comparado com a programação individual [FRO11]. Na indústria, Hulkko mediou a produtividade pela quantidade de linhas de código lógica desenvolvidas pelo tempo [HUL05]. Os resultados variaram muito entre os quatros projetos reais analisados no estudo [HUL05].

Em PP a colaboração é uma variável que teve efeito positivo entre equipes tanto em nível de indústria [VAN07a] quanto ao nível educacional [BIP07]. De acordo com Bipp a rotação entre os membros dos pares permitiu que todos que todos conhecessem as várias partes do projeto, facilitando a interação e colaboração entre a equipe [BIP07]. Na indústria, Vanhanen apontou que a PP ajudou no desempenho da equipe, a qual pode também desenvolver melhor outras práticas ágeis tais como TDD, padrões de código e integração contínua. PP ainda apresentou resultados positivos sobre a motivação da equipe [VAN07b]. Muller por meio de um experimento com alunos afirmou que PP deixava os programadores mais confortáveis e motivados nas sessões de codificação e isto ajudava no desempenho deles [MUL04]. Na indústria, uma survey aplicada aos programadores teve respostas positivas quanto a PP como facilitadora no processo de aprendizado [VAN07a].

Ensino

Dezesete estudos analisaram os efeitos que a prática de PP tem sobre o ensino de programação. A tabela 2.4 abaixo mostra os estudos, o tipo de amostra utilizada e as variáveis analisadas.

Tabela 2.4 – Lista de variáveis em torno de PP no ensino

Ensino					
Tipo de Amostra	Variáveis	Efeito Positivo	Efeito Negativo	Sem Efeito	Total de Artigos
Alunos	Confiança do programador	[HAN08], [HAN11], [MCD02], [RAM08], [SAL11], [WIE03]			6
	Notas dos alunos	[BRA08], {BRA11}, [BRE09], [MEN05], [WIE03], [WIL03]			6
	Qualidade do código	[MCD02], [CHI08], [HAN11], [RAM08]			4
	Desempenho do programador	[MCD02], [RAM08]			2
	Aprendizado	[CAR07], [NAG03]			2
	Diferentes tipos de personalidades			[SAL10], [SAL11]	2
	Produtividade	[CHI08], [NAG03]			2
	Motivação	[CHI08], [MCD02]			2
	Conversas off-topic		[SIS09]		1

No contexto educacional, alguns estudos [CHI08], [MCD02], avaliaram a qualidade do código em função das notas obtidas pelos alunos durante o curso como sendo um fator de melhoria. Han [HAN11] e Ramli [RAM08] avaliaram a qualidade do código a partir de algumas heurísticas que apresentaram bons resultados com PP, tais como: pouca quantidade de erros gramaticais e lógicos, mais tarefas finalizadas dentro de um espaço de tempo e uso de comentários e variáveis significantes ao longo do código desenvolvido pelos alunos.

A figura 2.11 apresenta a síntese dos resultados da tabela 2.4, ilustrando o tipo e a quantidade de estudos em torno dos efeitos das variáveis do ensino de PP.

Variáveis em torno do ensino de PP							
Confiança do programador		[RAM08]	[WIE03]	[HAN08]	[HAN11]	[SAL10]	[MCD02]
Notas dos alunos		[MEN06]	[WIL03]	[WIE03]	[BRA08]	[BRE08]	[BRA11]
Qualidade do código		[MCD02]	[CHI08]	[HAN11]	[RAM08]		
Desempenho do programador		[MCD02]	[RAM08]				
Aprendizado		[CAR07]	[NAG03]				
Diferentes tipos de personalidade	[SAL10], [SAL11]						
Produtividade		[CHI08]	[NAG03]				
Motivação		[CHI08]	[MCD02]				
Conversas off-topic	[SIS09]						
Legenda: Alunos	-1	0	1	2	3	4	5
	Negativo	Sem efeito	Positivo				

Figura 2.11 – Lista de variáveis em torno de PP no ensino

Quanto ao desempenho dos alunos, três estudos mostraram que PP ajudou os programadores a desenvolverem melhor as suas atividades. Ramli [RAM08] relata que os alunos obtiveram um melhor desempenho nas atividades em pares do que individualmente [RAM08]. McDowell apontou como desempenho duas métricas: a qualidade dos programas desenvolvidos e a capacidade dos estudantes aplicarem os conceitos ensinados durante o curso, as notas finais obtidas pelos alunos detalhada foi usada como parâmetro para esses critérios [MCD02].

Quatro artigos apresentaram que a PP é uma prática propícia para o aprendizado. Nagappan e Carver afirmam que PP auxilia na não evasão de alunos nos cursos de introdução a programação [CAR07], [NAG03]. Alguns estudos do tipo *survey* mostraram que os alunos ficaram mais motivados e tiveram mais satisfação ao usar a PP [CHI08], [MCD02]. Quanto à diferença de personalidade no uso de PP no ensino alguns estudos mostraram que não há um efeito significativo na PP e enfatizam que mais estudos precisam ser realizados [SAL09], [SAL10].

Outra variável importante que tem um efeito positivo no ensino em PP é a produtividade. Nagappan relatou que no curso de introdução de programação os alunos foram mais produtivos e menos frustrantes [NAG03]. A medida usada foi a nota dos alunos. Chigona também ratifica esses resultados por meio de um experimento, onde os alunos possuíram menos dúvidas e informaram por meio de uma *survey* que foram mais produtivos usando PP [CHI08].

PP também mostrou ser eficiente por meio do aumento das notas dos alunos. Seis estudos apontaram essa métrica positiva [BRA08], [BRA11], [BRE09], [MEN05], [WIE03], [WIL03]. Outros seis estudos apresentaram que PP aumentou a confiança do programador e todos eles foram aplicados em um contexto educacional em experimentos envolvendo alunos. Para medir a confiança os estudos aplicaram *surveys* após cursos que utilizavam PP [HAN08], [HAN11], [MCD02], [RAM08], [SAL11b], [WIE03]. Os resultados nestes estudos apontam que PP tem um efeito positivo na confiança ao desenvolver as atividades de programação. Nas universidades PP ainda mostrou ter um empecilho entre os alunos. Segundo Sison as conversas *off-topic* após a terceira rodada de programação aumentaram [SIS09].

O que se sabe sobre a utilização da programação em par distribuída?

Dos 22 artigos de PPD identificados nesta Revisão Sistemática, apenas 4 estudos trazem resultados empíricos em torno de variáveis de PPD.

Prática

Três artigos tratam de analisar PP sob a perspectiva de prática, envolvendo quatro variáveis. A tabela 2.5 mostra os estudos e as variáveis analisadas.

Tabela 2.5 – Lista de variáveis em torno da prática de PPD

Prática					
Tipo de Amostra	Variável	Efeito Positivo	Efeito Negativo	Sem Efeito	Total de Artigos
Alunos	Qualidade do código	[BAH02]	[CAN03], [CAN06]		3
	Produtividade	[BAH02]		[CAN06]	2
	Conhecimento		[CAN03]		2
	Comunicação	[BAH02]			1
Profissionais	Qualidade do código	[ROS10]			1
	Comunicação	[ROS10]			1
	Conhecimento	[ROS10]			1
	Distração		[ROS10]		1
	Cumprimento do papel		[ROS10]		1
	Conflito de objetivos		[ROS10]		1

A figura 2.12 apresenta a síntese dos resultados da tabela 2.4, ilustrando o tipo e a quantidade de estudos em torno dos efeitos das variáveis da prática de PPD.

Variáveis em torno da prática de PPD					
Qualidade de Código	[CAN03]	[CAN06]		[BAH02]	[ROS10]
Conhecimento				[CAN03]	[ROS10]
Produtividade			[BAH02]	[CAN06]	
Comunicação				[BAH02]	[ROS10]
Distração		[ROS10]			
Cumprimento do papel		[ROS10]			
Conflito de objetivos		[ROS10]			
Legenda: Alunos	-2	-1	0	1	2
Profissionais	Negativo	Sem efeito			Positivo

Figura 2.12 – Lista de variáveis em torno da prática de PPD

Em 2002, Baheti avaliou a efetividade da PPD por meio de um experimento, envolvendo alunos que estavam em times co-locados e outros que estavam em um ambiente distribuído, sendo que alguns times utilizavam programação em par e outros não [BAH02]. Duas métricas foram utilizadas: qualidade (nota dos alunos) e produtividade (Linhas de código por hora). Os resultados mostraram que em termos das duas métricas, a PPD foi equivalente estaticamente aos times que usavam PP de forma co-locada e os que não usavam a prática. O *feedback* dado pelos alunos participantes indicou que a PPD ajuda a promover o trabalho e a comunicação dentro de equipes distribuídas. Este experimento foi o primeiro a indicar que PPD é um método possível e eficiente para lidar com desenvolvimento distribuído de software.

Após esse experimento, outros experimentos foram conduzidos no sentido de se analisar a eficiência de PPD. Canfora realizou dois. O primeiro de duração de 11 semanas envolvia alunos que formavam oito pares co-locados e oito pares distribuídos [CAN03]. O experimento tinha por objetivo analisar o impacto da distribuição da PP segundo as métricas de produtividade (em termos de linhas de códigos) e qualidade do código (nota dos alunos). Os autores constataram uma hipótese rejeitada de que componentes dos pares tendiam a trabalhar só. Foram identificados quatro fatores para explicar este comportamento: o não estabelecimento de um protocolo de trabalho, o conflito de ideias entre os pares, problemas com o software de *chat* e os diferentes níveis de experiência

entre os pares. Com o intuito de coletar mais informações e minimizar o fenômeno da rejeição dos pares, o segundo experimento desenvolvido pelos autores foi uma réplica do primeiro [CAN06]. Na réplica a rejeição dos pares foi menor. Os resultados apontaram que a qualidade teve uma queda na programação em par distribuída por conta de questões como infraestrutura de colaboração e nenhuma evidência constatou que o esforço aumenta quando a prática é distribuída. Canfora apontou dois fatores para sucesso de PPD: uma comunicação apropriada e o suporte a colaboração.

O estudo de caso de um projeto piloto realizado por Rosen em duas filiais de uma empresa alemã é o único estudo de PPD que envolve profissionais da indústria identificado pela RSL [ROS10]. Como efeito positivo foram citados a melhora na comunicação, a transferência do conhecimento e a qualidade do código. Como efeito negativo foram citadas como variáveis: a distração, a falta de cumprimento do papel de PPD (seja de *driver* ou de *observer*) e o conflito de objetivos durante a sessão de PPD entre os desenvolvedores.

Ensino

O ensino de programação com a utilização de PPD foi abordado por apenas dois estudos. A tabela 2.6 e a figura 2.13 apresentam os estudos e as variáveis analisadas.

Tabela 2.6 – Lista de variáveis em torno de PPD no ensino

Ensino					
Tipo de Amostra	Variável	Efeito Positivo	Efeito Negativo	Sem Efeito	Total de Artigos
Alunos	Desempenho do programador	[HAN05], [ZAC11]			2
	Notas dos alunos	[HAN05]			1
	Produtividade	[ZAC11]			1
	Motivação	[ZAC11]			1

A figura 2.13 apresenta a síntese dos resultados da tabela 2.6, ilustrando o tipo e a quantidade de estudos em torno dos efeitos das variáveis do ensino de PPD.

Variáveis em torno do ensino de PPD					
Desempenho do Programador				[HAN05]	[ZAC11]
Notas dos alunos				[HAN05]	
Produtividade				[ZAC11]	
Motivação				[ZAC11]	
Legenda: Alunos	-2	-1	0	1	2
	Negativo	Sem efeito		Positivo	

Figura 2.13 – Lista de variáveis em torno de PPD no ensino

Hanks realizou um experimento com alunos a respeito de verificar o desempenho do uso de uma ferramenta de apoio a programação em par distribuída em um curso introdutório de programação [HAN05]. Os resultados mostraram que os alunos que realizaram PPD tiveram um desempenho tão bom quanto os alunos que estavam com pares co-locados, passando no curso com notas similares. Outra métrica utilizada foi o nível de confiança, que se mostrou estatisticamente semelhante.

Zacharis realizou um estudo com alunos para investigar a efetividade de PPD no desempenho dos alunos e a sua motivação em um curso de introdução de Java [ZAC1105]. Algo interessante sobre esse estudo é que a comparação foi feita com a programação individual e os resultados apontaram que os alunos que usaram PPD tiverem 50% menos defeitos, evidenciando a qualidade do código, e foram mais produtivos na métrica baseada em LOC/h [ZAC11]. O Desempenho também aumentou com as notas dos alunos. Os alunos também responderam um questionário ao final do curso, afirmando que PPD deu mais motivação e satisfação.

Em que condições a PP funciona?

Para responder a pergunta QP2 foram utilizadas a ficha de leitura e a categorização feita para a QP1. Após isso foram classificados os estudos onde PP obteve efeitos positivos.

A maior parte das evidências com efeito positivo para PP, isto é, condições onde a PP funciona se encontram nos estudos que analisam os efeitos da prática no desenvolvimento de software. Neste contexto, ainda há a predominância de estudos

envolvendo alunos para analisar as variáveis levantadas. Sob a perspectiva de ensino, relatos de experiência envolvendo cursos de introdução a programação [NAG03], [SIM08] e ensino de linguagens OO como Java [CHO07a], [HAN11] são exemplos de estudos que retratam resultados positivos em torno de PP.

Estudos envolvendo profissionais foram identificados bem poucos. Para qualidade do código, por exemplo, que foi apontada como a variável com efeito positivo mais citado nos estudos de PP tanto no enfoque quanto prática quanto ferramenta para ensino, somente dois trabalhos na indústria relataram como sendo positiva.

Nos estudos com foco no ensino, Bevan e Williams listaram algumas diretrizes para o bom funcionamento de PP neste contexto por meio de experimentos anteriores [BEV02], [WIL08]. Bevan ressalta algumas diretrizes para cursos de introdução a programação, como: instituir um padrão de código (visando diminuir a diferença de conhecimento entre os pares, pois nenhum dos dois pode dominar o estilo do código), fazer as seções de PP serem mandatórias (garantindo que um mínimo de tempo seja gasto entre os pares) [BEV02]. Já Williams enfatiza a importância de ter um *staff* de professores e monitores bem treinados para a prática e uma política clara para implantação de PP [WIL08].

Uma infraestrutura apropriada para PP se mostrou necessária, porém poucos estudos evidenciaram isso de forma clara. Na indústria, Chong apontou, por exemplo, que o controle do teclado é uma importante fator para o sucesso de PP e sugere que ambos os programadores tenham teclado para uma troca rápida de controle entre os pares [CHO07b]. Na educação, Williams corrobora para o controle do teclado e outros aspectos de infraestrutura no contexto educacional [WIL07]. De acordo com ela o laboratório deve ter dois monitores, dois mouses e dois teclados. Ela enfatiza que o ambiente deve ser preparado para uma troca mínima de cadeiras entre os papéis. Vanhanem destaca que uma sala específica para PP não atrapalha outros funcionários de uma empresa que não usa PP com todos os funcionários devido à intensa comunicação [VAN07b].

Quanto ao tamanho das equipes, não foi possível definir uma quantidade padrão de pessoas onde a prática foi implementada com efeitos positivos, esta quantidade variou bastante. No contexto educacional essa variação foi dada pela quantidade de alunos envolvidos nos cursos onde PP foi utilizada. Na indústria, Choi realizou um estudo de caso com uma equipe de 6 desenvolvedores [CHO07a], enquanto Vanhanem aplicou um estudo com 35 programadores de uma organização [VAN07b].

As evidências positivas de algumas condições onde PP funcionou está estreitamente relacionado ao contexto. No contexto da educação, por exemplo, as notas foram um parâmetro para indicar o quanto PP apóia pedagogicamente, além da confiança e desempenho que os alunos iam adquiridos ao longo dos cursos. Já na indústria tivemos respostas pontuais dos próprios programadores sobre os benefícios de PP por meio de surveys [BEG08] ou estudos de casos [CHO07a], [VAN07b], porém poucos estudos empíricos.

Muitos estudos utilizam métricas clássicas para avaliar a produtividade de PP, como LOC/h. Esse tipo de métricas em linguagens OO pode não ter um resultado efetivo para análise. Aconselha-se usar métricas que avaliem a produtividade sob a complexidade da tarefa como Pontos de Função ou métricas utilizadas em métodos ágeis como estórias por *sprint*.

A efetividade de PP ainda foi analisada por metodologias da engenharia de software baseada em evidência como revisões sistemáticas e meta-análises obtendo bons resultados. Salleh apontou que PP ainda é pouco usada para tarefas que não sejam de codificação e relata que os resultados sugerem que a produtividade em PP é igual ou melhor do que com a programação individual entre estudantes. Os resultados suportam que PP é uma importante ferramenta educacional [SAI11]. Brereton em sua revisão sistemática ratifica com seus resultados dizendo que PP ajuda no desempenho das notas dos alunos e para que eles passem de curso [BRE09]. Dyba através de uma meta-análise de estudos tanto da indústria quanto da educação, alerta que a complexidade da tarefa e do sistema é uma variável importante para a efetividade da prática [Dyb09].

Em que condições a PPD funciona?

Poucos estudos apresentaram evidências empíricas de PPD. A maioria dos estudos realizados envolvem alunos para avaliar PPD sob as perspectiva de ensino e como prática. Em geral esses estudos têm por objetivo analisar a efetividade da distribuição de equipes da prática, portanto os experimentos em geral compararam PP com PPD. Apenas um estudo de caso sobre a avaliação de PPD em um projeto piloto tratava com profissionais da indústria.

A principal condição para que PPD funcione é o uso de uma ferramenta específica. Ho [HO04] aponta que o uso de ferramentas específicas para PPD ajuda a torna a prática mais rápida. Canfora [CAN06] corrobora dizendo que uma ferramenta específica evita que o programador fique alternando entre diferentes ferramentas, gastando tempo. Este

ferramental tem que prover um bom canal de comunicação por meio de *chat* e divisão da área de trabalho entre os programadores. Rosen [ROS10] relata que o compartilhamento da tela e o vídeo são importantes funcionalidades em uma ferramenta específica de PPD para minimizar a distração entre os desenvolvedores.

Quanto ao tamanho da equipe, os relatos de experiência também não mostraram um padrão entre eles, semelhante ao que aconteceu com PP. Canfora [CAN06], por exemplo, não dá detalhes de quantos alunos estavam no experimento que ele realizou. Já Zachari [E95] aponta que 65 estudantes estiveram engajados, enquanto Baheti [BAH02] envolveu 132 estudantes. Rosen[E61] não cita quantos profissionais ao todo estiveram envolvidos no projeto piloto do estudo de caso, apenas relata que deles era experiente e os demais novos desenvolvedores.

PPD mostrou que pode ser tão efetiva em cursos de introdução a Programação e cursos de linguagem OO quanto PP, essa condição é semelhante à implantação de PP. Isto enfatiza que a PPD também é uma importante ferramenta pedagógica no ensino de programação. Nenhum estudo foi específico em relação ao tipo de distribuição que fora utilizada, isto é, se fora em dispersão local ou nacional. Portanto, não se teve como definir uma condição ideal para PPD e possíveis desafios como o idioma, não foram analisados.

As métricas utilizadas para avaliar PPD foram bem semelhantes as que foram utilizadas para avaliação de PP. Na produtividade a métrica mais comum foi a LOC/h [BAH02], [CAN06], [ZAC11] e na qualidade e desempenho as notas obtidas pelos [BAH02], [HAN05], [ZAC11].

Não há nenhuma revisão sistemática que envolva especificamente PPD. Bandukda [BAN10] realizou uma revisão da literatura que envolveu alguns dos estudos de PPD mapeados nesta revisão, sintetizando alguns dos seus resultados. Ele conclui que a PPD é uma alternativa viável para equipes distribuída para melhora da produtividade e da qualidade do produto.

2.5.7 Conclusões da RSL

Tanto os dados da análise quantitativa como da análise qualitativa apresentaram resultados importantes a respeito de evidências empíricas de PP e PPD. Esta seção tem por objetivo apresentar esses resultados, com base no tipo de análise conduzida.

Conclusões baseadas na análise quantitativa

Conclusão 1: Existe uma necessidade de estudos em PP em casos reais na indústria.

Poucos artigos de PP tratam de análise a prática em projetos reais na indústria com profissionais. Apenas seis artigos tratam de analisar PP nessa perspectiva. Estudos que focam em etnografias e observações também se mostram bons em extrair evidências empíricas e poderiam ser mais utilizados pelos pesquisadores. Deste tipo apenas três estudos usaram essa metodologia [CHO07b], [FRO11].

Conclusão 2: Existe uma boa oportunidade para mais estudos empíricos em PPD.

Em uma *survey* recente de 2011 publicado pela organização Version One Inc.[VER11] mostrou que 65% dos participantes pertencem a um time de desenvolvimento distribuído de software (DDS) [VER11], esse valor em dois anos teve um aumento em comparado com pesquisas anteriores da mesma organização que apontavam 58% dos participantes pertencem a uma equipe distribuída, este cenário corrobora para a tendência de DDS relatado por Prikladnicki [PRi03]. Ainda com a importância desse contexto, PPD se mostrou uma área com poucos estudos empíricos tanto sobre a perspectiva de prática como de ensino. A necessidade de estudos com profissionais da indústria ainda é maior, pois apenas um estudo de caso real nesta revisão retratou o uso de PPD. Ainda há necessidade de estudos mais consolidados em torno das variáveis e os respectivos efeitos que podem ser gerados em PPD.

Conclusão 3: Existe uma forte tendência para o uso e pesquisa dos efeitos empíricos de PP e PPD no ensino de programação.

Dezesete estudos de PP e três estudos de PPD buscaram analisar os efeitos no ensino. Ainda que se tenham vários efeitos de PP na educação demonstrado pelos estudos, pesquisas continuam a investigar *gaps* e evidências do seu uso no ensino, como trabalhos mais completos como o de Salleh [SAL11]. Podemos apontar dois fatores para isso os benefícios diretos de PP para os alunos sob os efeitos de variáveis como notas, aprendizado e motivação e a maneira mais fácil de implantar a prática sob o ambiente de um curso de programação, em forma de experimento ao invés de um caso real na indústria. Isto também demonstra a imaturidade da área em se expandir para ambientes menos controlados.

Conclusões baseadas na análise qualitativa

Conclusão 1: Existe a necessidade de se investigar os efeitos de PP e PPD na indústria com profissionais.

Como dito em uma conclusão na análise quantitativa há uma necessidade de investigar PP na indústria. Efeitos como qualidade, produtividade e esforço foram bem pouco analisados sob a perspectiva da prática com profissionais. Essas evidências

ajudariam a dar suporte aos benefícios que a prática pode oferecer. Em PPD, apenas um estudo de caso realizado na indústria buscou analisar os efeitos de PPD [SAL11], isto enfatiza ainda mais a necessidade de mais estudos neste contexto.

Conclusão 2: O nível dispersão das equipes foi pouco contextualizado em PPD.

Os experimentos envolvendo PPD como de Canfora [CAN06] e Hanks [HAN05] não contextualizaram o nível de dispersão dos alunos ao usar PPD. Esses experimentos apenas relatavam que os alunos estavam geograficamente distantes. Pelo contexto do trabalho se inferia que a distribuição se deu sempre de modo local, porém esses detalhes eram pouco explicitados. O enfoque maior dos estudos foi nos efeitos da prática de PPD, fornecendo-se poucos detalhes em nível de DDS, como a variável de dispersão das equipes. Esses resultados vão ao encontro a um recente estudo proposto por Šmite [SMI12], a qual corrobora dizendo que estudos que envolvem DDS não possuem uma definição de contexto e de termos padrão, sendo que muitos autores criam as suas próprias definições ao em torno do DDS.

Conclusão 3: Existe a oportunidade de explorar variáveis e seus efeitos em torno de DDS em PPD.

Poucos artigos de PPD levaram aspectos de DDS em consideração, um desses que poderia ser abordado é a comunicação. A maior parte dos experimentos foi feita com pessoas do mesmo país, portanto a comunicação sofreu pouca interferência. Seria interessante desenvolver uma pesquisa em torno de equipes de diferentes países com línguas primárias diferentes e avaliar o efeito que isso causa em PPD. Outra variável que poderia ser explorada é o fuso horário, verificando como seria tentar encaixar PPD em um fuso muito destoante, o que seria interessante dar prioridade neste caso em termos de atividade para o uso da prática.

Conclusão 4: Existe a necessidade de explorar mais diretrizes (boas práticas) para implantação de PP e PPD na indústria.

No contexto de ensino de programação alguns estudos [BEV02 WIL07] apresentaram diretrizes de como é possível utilizar PP com os alunos. Vannhanem [VAN07b], por meio de um estudo de caso também forneceu algumas diretrizes para implantar PP na indústria. Canfora [CAN06] reuniu lições aprendidas em um experimento de PPD. Essas boas práticas são essenciais para o uso efetivo de PP e PPD e precisam ser mais exploradas para facilitar a adoção das práticas ágeis no contexto de DDS.

2.5.8 Limitações desta Revisão Sistemática

Esta revisão sistemática possui algumas limitações. A primeira delas está relacionada ao fato de que os estudos selecionados foram todos de algumas bibliotecas *on line*. Estas bibliotecas foram escolhidas a partir de outras revisões sistemáticas desenvolvidas pela comunidade científica, e também de acordo com o conhecimento da área dos dois pesquisadores que a executaram e da disponibilidade de acesso das bibliotecas na universidade. É possível que outras bases não utilizadas nesse trabalho também contenham artigos de PP e PPD. Portanto, não é possível garantir a cobertura total de artigos sobre esse assunto.

Outra limitação desse trabalho está relacionada ao viés dos pesquisadores durante o processo de análise dos artigos. Para minimizar esta limitação, a revisão foi realizada por dois pesquisadores, conforme recomendações na literatura específica da área. O pesquisador com mais experiência validou as etapas da revisão sistemática como a classificação dos artigos, a extração dos dados e a síntese dos resultados provendo um *feedback* detalhado sobre os artefatos que foram desenvolvidos.

3 METODOLOGIA DE PESQUISA

Neste capítulo apresenta-se a metodologia de pesquisa utilizada neste estudo. Na seção 3.1 é apresentado o desenho de pesquisa e suas etapas. Na seção 3.2 são apresentados os aspectos metodológicos, a base metodológica do estudo de caso realizado é apresentada na seção 3.3.

Para o desenvolvimento de um conjunto de boas práticas para a PPD optou-se pela realização de um estudo com natureza empírica. Isso se fez necessário, pois a pesquisa tem a aplicação ao contexto real de práticas da indústria, com forte intervenção humana. Essa característica justificou a escolha de métodos de pesquisa com uma abordagem qualitativa.

O método primário dessa pesquisa foi o estudo de caso exploratório. Segundo Yin [YIN10], o estudo de caso exploratório auxilia em áreas onde há pouca literatura sobre um determinado tópico, então uma instância do contexto real é investigada, auxiliando na identificação de tópicos para pesquisa. Utilizou-se como instrumento de pesquisa um roteiro para entrevista semi-estruturada com questões abertas.

3.1 Desenho e etapas da pesquisa

O desenho de pesquisa contempla as etapas necessárias para se alcançar o objetivo do estudo. Nesta pesquisa, o desenho de pesquisa (figura 3.1) foi constituído por três etapas e cinco fases, descritas:

Etapa 1: esta etapa é exploratória, e possui apenas uma fase (F1). Nesta fase buscou-se estudar o referencial teórico, envolvendo inicialmente os conteúdos de engenharia de software, métodos ágeis para desenvolvimento de software e DDS. Durante esta etapa foram estudados os conceitos de métodos ágeis de desenvolvimento de software e DDS. Parte desse estudo foi realizado durante os trabalhos da disciplina de Processo de Desenvolvimento de Software ofertada no segundo semestre de 2011 pelo Programa de Pós-Graduação em Ciência da Computação da PUCRS. Outro passo foi a elaboração de uma monografia que tinha por objetivo buscar oportunidades de pesquisa na área de métodos ágeis. A estratégia adotada foi à análise de cinco artigos científicos de maior impacto nas citações da base do Google Scholar¹ em Dezembro de 2011. Os resultados da análise apontaram para os seguintes temas: estudo de atributos de desempenho do uso do método *Scrum*, customização entre diferentes métodos ágeis e o tema desta pesquisa, programação em par com equipes distribuídas. Aliado aos

¹<http://scholar.google.com.br/>

resultados da monografia, a vivência como bolsista de mestrado no ambiente de desenvolvimento da HP *Enterprise Services* colaborou para que os conceitos fossem consolidados não apenas na teoria, colaborando também na identificação da questão de pesquisa também sob a perspectiva da indústria.

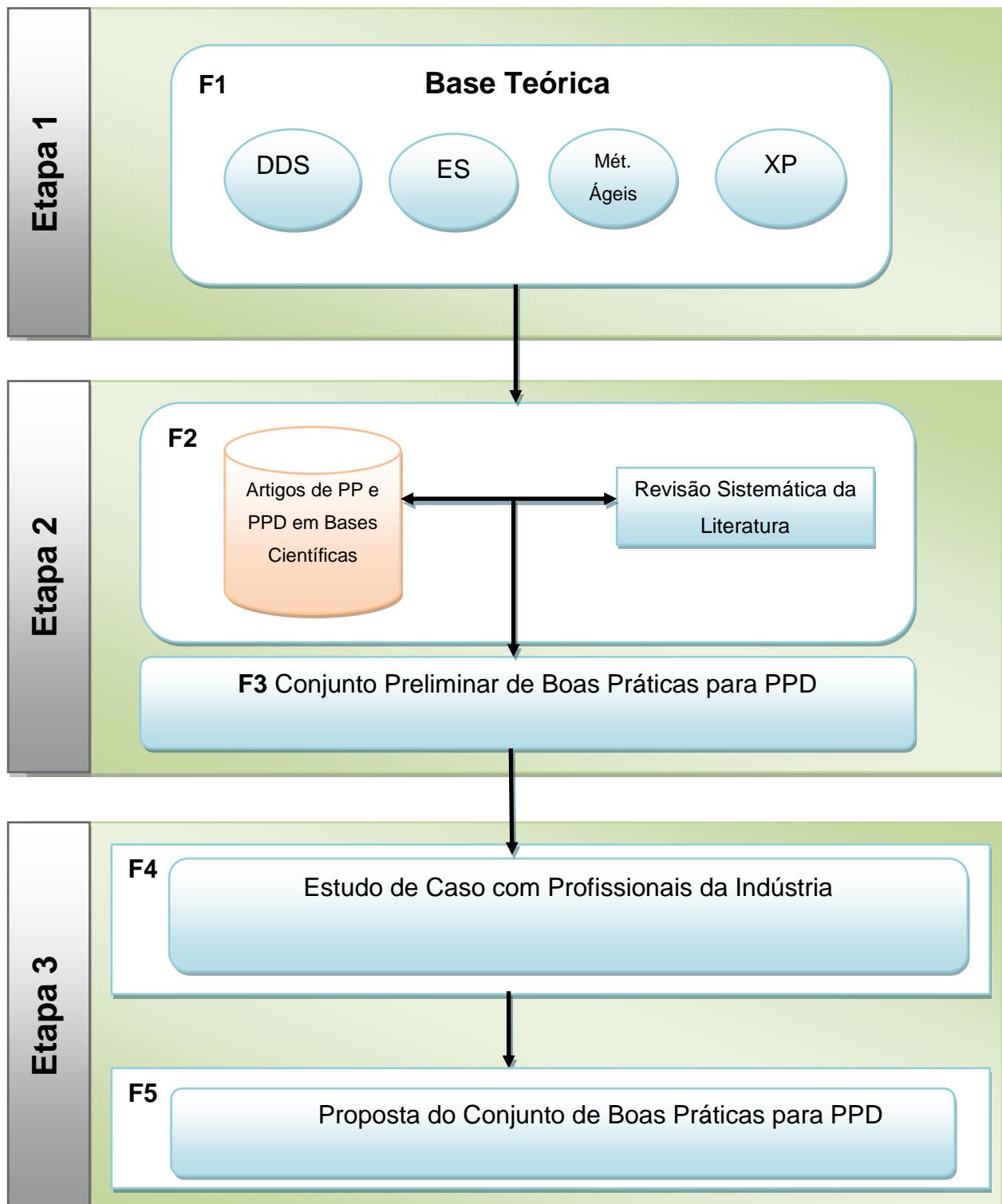


Figura 3.1 – Desenho de Pesquisa

Etapa 2: uma vez que a questão de pesquisa foi desenvolvida, na segunda etapa planejou-se a segunda fase (F2) que consistia na execução de um estudo secundário, revisão sistemática da literatura de PPD. O objetivo da revisão sistemática foi de aprofundar o entendimento das práticas de PP e PPD com os estudos existentes, ampliando a revisão bibliográfica inicial. A fase 3 (F3) desta mesma etapa tinha por objetivo analisar os resultados da RSL visando elaborar um conjunto preliminar de boas práticas de PPD. Durante o processo de leitura dos artigos científicos, foram identificadas estratégias tanto em nível de PP quanto em PPD, identificando a consequência na utilização e o indicador de melhoria. A opção de pesquisar também os estudos de PP se deu pelo fato de tentar avaliar se as boas práticas para PP também apoiariam a PPD, esta avaliação consistia na análise dos resultados dos múltiplos estudos de caso com o conjunto de boas práticas de PP identificado na RSL. A RSL permitiu a identificação dos trabalhos já existentes sobre o tema e a extração de resultados e lições aprendidas a respeito da PP e PPD.

Etapa 3: com o conjunto de boas práticas preliminar de PPD elaborado, nesta etapa planejou-se o desenvolvimento de múltiplos estudos de caso, de forma a consolidar e fornecer mais subsídios para o conjunto preliminar de boas práticas. Na fase 4 (F4) dessa etapa envolveu entrevistas com profissionais de diferentes empresas que possuíam experiência em projetos com PPD. As entrevistas avaliavam os resultados obtidos na segunda fase da etapa 2 e prospectaram novos resultados com base na experiência profissional dos entrevistados. A última fase (F5) da pesquisa consistiu na consolidação das boas práticas oriundas do conjunto preliminar obtido na RSL e das análises das entrevistas dos múltiplos estudos de caso.

3.2 Aspectos metodológicos

De forma a garantir e ampliar a validade deste estudo, um rigoroso processo de pesquisa foi planejado. Nesta pesquisa, a definição e utilização de protocolos para desenvolvimento e formalização dos múltiplos estudos tiveram por objetivo sistematizar as tarefas de observação e análise, aumentando a confiabilidade da pesquisa. A revisão sistemática foi planejada utilizando um protocolo de acordo com Kitchenham[KIT07]. Já os múltiplos estudos de caso, além de um protocolo formal, passaram por uma validação de face e conteúdo e um pré-teste com o objetivo de garantir a integridade dos resultados.

Quanto à generalização dos resultados, os estudos de caso não possibilitam a generalização estatística [YIN10]. Desta forma, o pesquisador procura um conjunto peculiar de resultados (casos), onde seja possível gerar proposições teóricas que seriam

aplicadas a outros contextos. Yin [YIN05] classifica isso de generalização analítica, a qual pode ser usada tanto para apenas um caso quanto múltiplos casos.

O viés de todo o processo de pesquisa foi minimizado com a avaliação das etapas de pesquisa com um pesquisador experiente (o professor orientador). Na revisão sistemática, a avaliação ocorreu principalmente no protocolo, na ficha de leitura dos artigos e na análise dos resultados. No estudo de caso, o protocolo da coleta dos dados e a análise dos resultados também contaram com esta avaliação.

3.3 Base metodológica do estudo de caso

O estudo de caso é particularmente adequado ao exame exploratório dos fenômenos ainda pouco estudados e que precisam ser investigados em seu ambiente de ocorrência [YIN05]. A utilização do estudo de caso é possível quando se tem por objetivo aprender sobre o estado da arte e gerar novas teorias apoiadas na prática, entender a natureza e complexidade do processo, enquanto este acontece, e trazer novos fatos e informações, evidenciados durante a execução de processo estudado [PRI03].

O estudo de caso foi desenvolvido com profissionais de várias organizações que possuíam experiência com PPD. O objetivo deste estudo em específico foi entender a execução de PPD no contexto de indústria. Buscou-se um entendimento dos resultados encontrados previamente na literatura apontada pela RSL, tanto de PP quanto de PPD, verificando-se estes seriam comprovados pela experiência dos profissionais com PPD.

3.3.1 Seleção das organizações e unidade de análise

A unidade de análise do estudo foi definida como sendo projetos de desenvolvimento de software envolvidos com o uso de PPD. Desta forma, foram escolhidos, por conveniência, profissionais que já tinham tido alguma experiência em algum projeto de desenvolvimento de software com PPD ou os que estavam em participando de um projeto que estava utilizando PPD.

Os profissionais selecionados pertenciam a nove organizações diferentes e todas colaboraram no processo de coleta de dados por meio das entrevistas. No Capítulo 4 apresentam-se detalhadamente os resultados encontrados em cada uma dos casos estudados nesta etapa.

3.3.2 Fonte dos dados e seleção dos participantes

A coleta de dados foi constituída por fontes primárias. As fontes primárias foram constituídas de entrevistas. Foram realizadas 14 entrevistas semiestruturadas individuais

em profundidade. Partiu-se de um roteiro básico com questões formuladas aos entrevistados e adequadas conforme seu desenvolvimento.

O critério inicial para a definição dos entrevistados centrou-se na unidade de análise e nos objetivos do estudo. Neste sentido, a população envolvida constituía-se de colaboradores que possuíam o perfil de desenvolvedores de software ou que já haviam tido alguma experiência neste sentido. O instrumento de coleta de dados consistiu de um roteiro para entrevista semi-estruturada (Apêndice C). As entrevistas foram organizadas para identificar boas práticas, benefícios e desafios de PPD.

3.3.3 Análise dos dados

A técnica utilizada para a análise de dados foi a de análise de conteúdos [MOZ10]. Desta forma, todas as entrevistas foram gravadas, transcritas e analisadas posteriormente. Após as transcrições, os dados foram preparados, uma leitura cuidadosa foi realizada, de modo a buscar a familiarização do pesquisador com os dados antes de iniciar a codificação. Para cada pergunta da entrevista foram identificadas categorias onde os dados foram codificados. Este processo foi conduzido pelo pesquisador e depois consolidado com o orientador, avaliando o conjunto de categorias a serem considerados.

3.3.4 Fases e operacionalização do estudo de caso

A pesquisa foi desenvolvida com 14 profissionais de 9 organizações. Em um primeiro momento, buscaram-se indicações com especialistas em métodos ágeis para conhecer pessoas e organizações que tinha experiência em PPD. Após entrar em contato com cada um dos indicados, todos os participantes dispuseram cerca de uma hora de tempo para as entrevistas que ocorreu tanto de forma presencial quanto a distância (via skype), quando os participantes estavam fisicamente distantes em outras cidades ou outros países.

Um questionário semi-estruturado foi utilizado como instrumento de coleta. Este questionário foi desenvolvido tomando-se por base um roteiro inicial de questões, a partir dos resultados da RSL executada e representada pelos protocolos de pesquisa desenvolvidos para os estudos de caso (Apêndice C).

A validação de face e conteúdo do protocolo de pesquisa foi realizada por uma profissional da indústria com experiência acadêmica (mestre). A partir disso foi executado um pré-teste, com um pesquisador que é aluno do mestrado da PUCRS. Após sua aplicação foi possível identificar o que poderia ser melhorar e adaptar as perguntas do

protocolo para se adequar ao objetivo da pesquisa. Foram realizadas iterações sucessivas até gerar uma versão estável do roteiro. Uma vez que parte das entrevistas seria realizada em inglês e parte em português, adotou-se como uma prática necessária a validação de face e conteúdo por pesquisadores nos dois idiomas.

A análise de conteúdo seguiu várias etapas, que iniciaram pela definição do universo estudado, delimitando o que estaria envolvido. Em seguida, iniciou-se a categorização, representando tópicos significativos em função das quais o conteúdo foi classificado. Os resultados das transcrições foram categorizados e, por fim, analisados.

4 RESULTADOS DO ESTUDO DE CASO

Este capítulo apresenta os resultados dos estudos de caso realizados. Nas seções 4.1 e 4.2 são caracterizados os projetos analisados e os respondentes. Na seção 4.3 os resultados dos estudos de caso são apresentados de acordo com a categorização realizada a partir da análise de conteúdo. Por fim, a seção 4.4 apresenta as lições aprendidas dos estudos de caso.

4.1 Caracterização dos projetos analisados

Esta pesquisa envolveu 10 projetos de desenvolvimento de software que utilizaram PPD de 8 organizações diferentes. A tabela 4.1, apresentada a seguir, contém um resumo das informações dos projetos analisados.

Tabela 4.1 – Resumo dos Projetos Analisados

Empresa	Projeto	Entrevistados	Nível de Distribuição [PRI03]	Idioma da Equipe	Frequência de Utilização de PPD	Experiência Anterior
Empresa A	Projeto 1	4	Global (Brasil, Índia e EUA)	Inglês	Pontual	Apenas PP
Empresa A	Projeto 2	2	Global (Brasil, Índia e EUA)	Inglês	Pontual	Apenas PP
Empresa B	Projeto 3	1	Global (Brasil, Índia, Rússia e China)	Inglês	Pontual	Sem experiência
Empresa B	Projeto 4	1	Global (Índia, China, Brasil)	Inglês	Pontual	Sem experiência
Empresa C	Projeto 5	1	Nacional (Brasil)	Português	Pontual	Apenas PP
Empresa D	Projeto 6	1	Nacional (EUA)	Inglês	Integral	PP e PPD
Empresa E	Projeto 7	1	Global (Macedônia, África do Sul)	Inglês	Pontual	Sem experiência
Empresa F	Projeto 8	1	Continental (Brasil, EUA)	Inglês	Pontual	Sem experiência
Empresa G	Projeto 9	1	Continental (Polônia, UK)	Polonês	Pontual	Sem experiência
Empresa H	Projeto 10	1	Nacional (Brasil)	Português	Pontual	PP e PPD

Apresenta-se a seguir as principais informações sobre os projetos destacados na tabela 4.1, sendo detalhadas as práticas da PPD empregadas em cada um deles.

4.1.1 Projeto 1

O projeto 1 faz parte da Empresa A, uma multinacional norte-americana que tem como característica o desenvolvimento ágil de software. O projeto 1 tinha como contexto

um sistema bancário com 20 colaboradores, sendo 8 estão no Brasil, 4 na Índia e 8 nos Estados Unidos. A diferença de fuso horário da equipe brasileira para a americana varia de 4 a 6 horas, já para a equipe indiana chega a 8,5 horas de diferença. Desta forma, a PPD é utilizada de maneira pontual, quando necessária para tarefas críticas. A equipe já possuía experiência com PP, antes da adoção de PPD. O idioma utilizado no projeto é o inglês.

A entrevista foi realizada presencialmente em Porto Alegre com 3 pessoas separadamente. Um quarto colaborador participou da pesquisa individualmente via Skype.

4.1.2 Projeto 2

O projeto 2 também pertence a empresa A e consistia no desenvolvimento de uma aplicação web para vendas de varejo que envolvia também as três unidades: Brasil, Índia e Estados Unidos. O projeto contava com a participação de 6 colaboradores, sendo dois colaboradores em cada unidade. Semelhante ao projeto anterior, a equipe já possuía experiência com PP e a utilização de PPD foi feita de forma pontual devido a diferença do fuso-horário. Dois colaboradores de Porto Alegre participaram presencialmente da entrevista em momentos diferentes.

4.1.3 Projeto 3

O projeto 3 faz parte da empresa B, uma multinacional norte-americana de grande porte. O projeto tinha como contexto a manutenção de um módulo de ERP da organização e possuía equipes do Brasil, Índia, Rússia e China. O projeto contava com aproximadamente 15 colaboradores em cada unidade. Neste projeto devido à variação do fuso-horário (que chegava a ser de 8,5 horas do Brasil para a Índia), a PPD foi usada pontualmente em sessões curtas. A equipe não possuía experiência alguma tanto com PP quanto com PPD. Um colaborador participou presencialmente da entrevista.

4.1.4 Projeto 4

O projeto 4 também pertence a empresa B e envolvia as unidades da Índia, China e Brasil e possuía o mesmo contexto do projeto 3, entretanto a manutenção era em outro módulo do ERP da organização. A equipe não possuía experiência com PP nem com PPD. Deste projeto, um colaborador foi entrevistado presencialmente.

4.1.5 Projeto 5

O projeto 5 faz parte da empresa C. Esta é uma empresa brasileira de médio porte com matriz em Santa Catarina. O projeto 5 desta pesquisa era o desenvolvimento de um sistema bancário que contava com uma equipes 2 colaboradores em Florianópolis e 4 colaboradores em Curitiba. PPD foi usado pontualmente para tarefas críticas. O idioma utilizado foi o português. A equipe já possuía experiência com PP. Um colaborador foi entrevistado via Skype.

4.1.6 Projeto 6

O projeto 6 faz parte da empresa D, a qual é uma empresa norte-americana de porte médio. O projeto consistia em uma aplicação web de compras coletivas. A distribuição das equipes se dava em duas cidades americanas, uma colaborador em Nova York e outros 4 colaboradores em Atlanta. Não há diferença de fuso-horário entre as equipes e o idioma utilizado é o inglês. Este é o único projeto analisado que utiliza PPD em tempo integral. Os membros da equipe já possuíam experiência com PP e PPD. Um colaborador foi entrevistado via Skype.

4.1.7 Projeto 7

O projeto 7 foi desenvolvido na empresa E, uma empresa de porte médio localizada na Macedônia. O projeto consistia no desenvolvimento de um sistema governamental. As equipes estavam distribuídas na África do Sul e na Macedônia e possuíam o mesmo fuso-horário. Cada unidade contava aproximadamente com 6 colaboradores. O idioma utilizado foi o inglês. PPD foi usado pontualmente em algumas tarefas. Os membros da equipe não tinham experiência com PPD. Um colaborador foi entrevistado via Skype.

4.1.8 Projeto 8

O projeto 8 pertence a empresa F, uma organização norte-americana de porte médio. O projeto consistia em uma aplicação de arquivamento de emails. A distribuição de equipes se deu entre 1 colaborador Brasil (Campinas) e outros 3 colaboradores do Estados Unidos (Nova York). A PPD foi usada para algumas tarefas e a variação do fuso-horário entre as equipes chegava a 3 horas. Os membros não tinham experiência com PPD. A entrevista ocorreu por Skype com um colaborador.

4.1.9 Projeto 9

A empresa G é uma empresa polonesa de porte pequeno, o projeto 9 consistia em um aplicativo móvel para uma *start up*. A distribuição das equipes se deu entre a Polônia e a Inglaterra, o idioma utilizado pela equipe foi o polonês, pois os membros da equipe estavam temporariamente na Inglaterra. O projeto contou com aproximadamente 5 colaboradores em cada unidade. A equipe não tinha experiência com PPD. Um colaborador foi entrevistado via Skype.

4.1.10 Projeto 10

A empresa H, uma organização brasileira de porte pequeno desenvolveu o projeto 10 que consistia em um sistema bancário. A distribuição das equipes se deu entre dois colaboradores de Porto Alegre e seis colaboradores de São Paulo. A equipe já tinha experiência com PP e PPD. A prática era usada de forma pontual para algumas tarefas. Um colaborador participou presencialmente da entrevista.

4.2 Caracterização dos respondentes e sua participação

As entrevistas foram realizadas com 14 profissionais que foram selecionados em função da sua experiência em um projeto de desenvolvimento de software que faz uso ou utilizou PPD.

Em relação à experiência dos entrevistados, todos possuíam pelo menos 2 anos de experiência com desenvolvimento de software, sendo o tempo médio de experiência de 8,2 anos. Quanto à experiência com DDS, o tempo médio dos entrevistados é de 3,7 anos. O tempo médio de experiência com métodos ágeis é de 5,4 anos. Todos os entrevistados tinham experiência com PP, sendo o tempo médio de 3,8 anos. Já a experiência com PPD registrou o tempo médio de 2 anos, sendo o menor informado de 6 meses.

Com relação ao nível de formação, apenas um entrevistado não possui graduação superior completo em TI, sua graduação é em bacharelado em Administração, porém sua atuação é com desenvolvimento de software. Dois entrevistados possuem mestrado em Ciência da Computação. Todos exerceram o papel de desenvolvedor de software, porém atualmente dois são Consultores de Software e um é proprietário de uma empresa de TI.

4.3 Resultados do estudo de caso

A categorização e a análise dos resultados dos conteúdos dos múltiplos de estudos de caso permitiram traduzir a realidade estudada e o seu impacto nos objetivos desta pesquisa. A seguir apresentam-se os elementos analisados e as categorias obtidas.

4.3.1 Aspectos de DDS

As questões dessa categoria buscavam identificar a percepção e a experiência em termos de benefícios e desafios dos respondentes em relação ao DDS. Pelos entrevistados foram citados benefícios de DDS em relação à maior flexibilidade (a possibilidade de se trabalhar em *home office*), a possibilidade de se trabalhar com bons desenvolvedores de outras nacionalidades e a retenção de bons recursos humanos de diferentes localidades trabalhando juntos.

Alguns trechos das entrevistas desenvolvidas permitem ilustrar estes resultados, como, por exemplo, este de um desenvolvedor do projeto 5:

“Com o DDS conseguimos trabalhar com desenvolvedores competentes de diferentes locais do país, retendo profissionais que tornam o nosso time mais forte. Outro benefício é a flexibilidade, às vezes necessitamos viajar para a cidade onde estão nossos clientes, a outra parte do time fica no escritório, mas o ritmo de trabalho continua o mesmo.”

Os entrevistados citaram como desafio de DDS: a comunicação entre o time, o idioma, o conhecimento técnico do projeto e o fuso-horário. As soluções citadas para os desafios da DDS foram: constantes reuniões com o time, uma boa infraestrutura de comunicação, treinamentos e a utilização de monitores e televisores para mostrar o ambiente de trabalho. A seguir, um trecho de um desenvolvedor do projeto 1:

“Os principais desafios de DDS que o nosso time enfrenta é em relação à comunicação, a intranet as vezes fica sobrecarregada, o que atrapalha o trabalho. Outro problema que enfrentamos é a diferença de conhecimento do projeto e técnico. Para resolver isso, fazemos diárias para discutir o projeto e promovemos umas palestras sobre tecnologia para a equipe, assim conseguimos minimizar a diferença.”

4.3.2 Adoção de PPD

Foram citados como principais fatores de adoção da PPD: difundir conhecimento de negócio e técnico entre as equipes distribuídas, resolver tarefas críticas, melhorar a

comunicação dos desenvolvedores, minimizar a distância entre os desenvolvedores foram os principais fatores que levaram a organização a usar PPD.

O momento da adoção de PPD, segundo os respondentes ocorreu tanto no início como durante o projeto. Neste trecho, o desenvolvedor da empresa F comentou a adoção de PPD:

"Utilizamos PPD durante o todo projeto, pois ajudava a melhorar a comunicação entre os desenvolvedores, ao invés de utilizar apenas email."

O colaborador entrevistado no projeto 3 comentou:

"Evitávamos utilizar PPD devido à diferença no fuso-horário para China, optamos por utilizar sessões curtas de 2 horas apenas em tarefas complicadas, onde o desenvolvedor não estava conseguindo chegar à solução do problema ou quando era necessário compartilhar alguma informação de negócio."

4.3.3 Variáveis de PPD

Nesta categoria, as perguntas tinham por objetivo avaliar os resultados da RSL quanto aos efeitos sobre algumas variáveis no desenvolvimento de software. Em relação à qualidade do código todos os respondentes relataram que o uso de PPD proporciona um efeito positivo nessa variável. O entrevistado do projeto 10 afirmou:

"A PPD teve um efeito positivo na qualidade do código, com frequência o programador que pareava comigo me alertava de um erro que estava cometendo e o produto que entregamos teve uma margem de defeitos baixa. Acredito também que para PPD sempre ajudar na qualidade, é necessário ter o move people around, isto é, ter uma constante troca entre os pares, caso contrário o par fica viciado no erro do outro par."

Em relação à produtividade, alguns respondentes citaram que PPD ajudou no melhor andamento das atividades, como relata um desenvolvedor do projeto 6:

"PPD ajuda muito no fluxo das atividades, antes quando usávamos apenas e-mails, perdíamos um pouco de tempo, principalmente na resolução dos problemas".

O entrevistado do projeto 2 afirmou:

"Tivemos bastantes dificuldades na implantação de PPD, principalmente no uso de ferramentas e no nosso link da internet, isso deixava a prática lenta. Acredito que tenha sido um efeito negativo na produtividade."

A melhoria da comunicação foi apontada como um efeito positivo de PPD pelos entrevistados. O respondente do projeto 5 relatou:

"PPD ajuda a criar o conceito de time, ao invés de trazer todos para a mesma cidade para fazer uma integração, podes trabalhar remoto e evitar os custos de viagens. Além disso, a comunicação pareada é melhor do que apenas trocar email."

Quanto à diferença de conhecimento, todos os respondentes relataram que PPD ajuda para que essa diferença não seja problema. Um desenvolvedor do projeto 3 afirmou:

"PPD colabora para diminuir o problema da diferença do conhecimento tanto na parte técnica quanto de negócio. Quando entrava um colaborador novo na equipe, nós o colocávamos para parear com um desenvolvedor que já estava bastante tempo no projeto, isso ajudava ele aprender mais rápido e também ajudava o cara mais experiente a encontrar os erros no código, por exemplo."

4.3.4 Características de PPD

Essa categoria tinha por objetivo avaliar algumas características da aplicação de PPD nos projetos analisados, bem como avaliar algumas boas práticas do conjunto preliminar levantados na RSL.

Quanto ao uso de uma diretriz organizacional (*guideline*) nenhum dos entrevistados utilizava este artefato nos projetos. Um desenvolvedor do projeto 1 relatou:

"Aplicamos a PPD de forma empírica, sem nenhum manual, o que fizemos uma vez foi um treinamento para equipe sobre a prática, mas não chegamos a desenvolver um artefato que apóie a implantação da prática. Acredito que não seja necessário."

Quanto à infraestrutura e o uso de ferramentas, os projetos utilizaram diferentes tipos de abordagem. A tabela 4.2 apresenta as configurações usadas pelos projetos analisados.

Tabela 4.2 – Configuração dos Projetos Analisados

Projeto	Infraestrutura	Ferramenta	Ferramenta Específica de PPD
Projeto 1	Sala de Reunião com Monitor/TV e Webcam	VNC/ Skype	Não utilizou
Projeto 2	Sala de Reunião com Monitor/TV e Webcam	Skype/ Saros	Saros
Projeto 3	Própria Estação de Trabalho	Microsoft Communicator/ Live Meeting	Não utilizou
Projeto 4	Própria Estação de Trabalho	Skype	Não utilizou
Projeto 5	Própria Estação de Trabalho	Skype	Não utilizou
Projeto 6	<i>Home Office</i>	Skype/Tmux/IChat	Não utilizou
Projeto 7	Própria Estação de Trabalho	Skype/Tmux	Não utilizou
Projeto 8	<i>Home Office</i>	VNC	Não utilizou
Projeto 9	Própria Estação de Trabalho	Skype	Não utilizou
Projeto 10	Própria Estação de Trabalho	Skype	Não utilizou

Apenas o projeto 2 utilizou uma ferramenta específica para PPD, outros 8 projetos utilizam Skype para PPD. Dois projetos realizam PPD através de *Home Office*, outros dois utilizam a sala de reunião da empresa e o restante utiliza a própria estação de trabalho dentro da empresa.

Quanto à presença de um *coach* que facilitasse a prática, três projetos (3, 6 e 10) possuíam esse papel. O desenvolvedor do projeto 6 relatou:

“Eu exerço esse papel, respondo todas as dúvidas da equipe em relação à prática e promovo-a na organização. Às vezes, promovemos treinamento sobre a implantação de PPD e estimulamos o feedback dos colaboradores constantemente. Esse papel tem estimulado o time na adoção da prática e tem apoiado nos desafios que temos percebido.”

O critério de formação dos pares foi diferente para cada projeto, a tabela 4.3 apresenta essas informações.

Tabela 4.3 – Características dos Pares entre os projetos

Projeto	Critério de Formação dos Pares	Responsável pela Formação dos Pares
Projeto 1	Tarefa	Time
Projeto 2	Negócio	Time
Projeto 3	Negócio	Time
Projeto 4	Tarefa	Time
Projeto 5	Negócio	Time
Projeto 6	Experiência	Coach
Projeto 7	Tarefa	Time
Projeto 8	Tarefa	Time
Projeto 9	Tarefa	Time
Projeto 10	Tarefa	Time

Na maioria dos projetos quem é responsável pela formação dos pares é o próprio time, apenas no projeto 6 que o *Coach* possui esse papel. Quanto ao critério da formação dos pares, três projetos formam os pares de acordo com o conhecimento de negócio dos desenvolvedores, enquanto seis projetos escolhem pares de acordo com a tarefa a ser realizada. O Projeto 6 seleciona os pares de acordo com o nível de experiência, independente da tarefa, o colaborador deste projeto relatou:

“O critério de formação dos pares é o nível de experiência, isso possibilita o maior compartilhamento de conhecimento entre a equipe, além da tarefa não pertencer a único par ou grupo específico.”

O desenvolvedor do projeto 3 que utilizava como critério de formação dos pares o conhecimento de negócio afirmou:

“Um dos principais desafios que percebíamos em PPD era a diferença de conhecimento em relação ao negócio do projeto. Os pares, então eram formados por um desenvolvedor com muito conhecimento do projeto e do negócio com outro desenvolvedor em aprendizado, novo na equipe.”

O desenvolvedor do projeto 10 que utilizava como critério de formação dos pares a tarefa comentou:

“Utilizávamos como critério de formação dos pares a tarefa, cada requisito formava um par diferente e ao término de desenvolvimento destes requisitos

novos pares eram formados. Isso possibilitava que a equipe não ficasse apenas em módulo do sistema, os membros da equipe pareavam em diferentes tipos de tarefa.”

Todos os entrevistados relataram que acreditam que com a PPD a diferença de experiência e conhecimento não é um empecilho. O entrevistado do projeto 3, relatou:

“Não acredito que seja um problema, acredito que seja o ideal. Quando pariei com um desenvolvedor experiente consegui aprender muito da experiência da pessoa e a distribuição não fez diferença alguma.”

4.3.5 Benefícios de PPD

Esta categoria buscava identificar os benefícios de PPD, os resultados da RSL embasaram algumas perguntas. Quanto ao compartilhamento de conhecimento, todos os entrevistados concordaram que este é um dos principais benefícios de PPD. Em relação à diminuição no tempo de execução de uma tarefa, o respondente do projeto 4 afirmou:

“A gente utilizava PPD de forma reativa no projeto, então a PPD ajudava a acelerar determinadas tarefas que não conseguíamos resolver sozinhos.”

Quanto ao esforço, todos os entrevistados relataram que PPD não reduz o esforço e sim aumento. O respondente do projeto 3, relatou:

“A PPD exige mais esforço, pois requer mais foco e menos distração. A colaboração é mais intensa. Acredito que com sessões mais curtas, esse problema possa ser diminuído.”

O desenvolvedor do projeto 6 que utilizava PPD de maneira integral, comentou:

“PPD não diminui o esforço de trabalho, após o dia de trabalho ficávamos muito cansados. Entretanto, utilizávamos algumas estratégias como: pausas de 30 minutos e a troca de pares entre os turnos.”

Quanto à motivação como um benefício de PPD, o respondente do projeto 9 relatou:

“A motivação não é da prática de PPD em si e sim da equipe. Acredito também que se a organização incentiva métodos ágeis, a equipe estará motivada em aceitar as práticas ágeis, como PPD.”

Alguns outros benefícios foram citados pelos entrevistados, tais como:

- Relacionamento do time: foi percebida uma melhor comunicação com a adoção de PPD, os integrantes da equipe passaram a entender mais do projeto e a colaboração mútua que a prática exige possibilitou que a interação fosse maior entre todos do projeto.

- Comprometimento: a PPD é uma prática que requer menos distrações, isso possibilitou que a equipe se empenhasse mais. A responsabilidade que se estabelece com um par na definição de um horário para sessões contribuiu também para o comprometimento da equipe.
- Solucionar Tarefas Críticas: o uso de PPD possibilitou que a equipe de desenvolvimento ajudasse a resolução de tarefas críticas.

Com base nas respostas obtidas, conclui-se a partir da tabela 4.4 que os principais benefícios de PPD são:

Tabela 4.4 – Benefícios da PPD

Benefícios da Programação em Par Distribuída
Comprometimento
Relacionamento do time
Solucionar tarefas críticas
Tempo de execução da tarefa
Compartilhamento de conhecimento

4.3.6 Desafios de PPD

Esta categoria tinha objetivo entender os desafios de PPD. Quanto à comunicação, os principais desafios relatados se deram em torno da infraestrutura. Como rela o respondente do projeto 3:

“A conexão da internet da Índia não era tão boa e isso dava um delay quando fazíamos PPD. Em determinados momentos, o desenvolvedor narrava o que estava fazendo, mas o cursor do mouse ou do teclado ainda não tinham executado aquela ação.”

Outro desafio enfrentado foi o idioma de equipes de diferentes países. O respondente do projeto 3 relata isso:

“Às vezes tínhamos um problema de comunicação em relação ao idioma, o sotaque do inglês de outros países como China e Índia tornavam a prática um pouco mais difícil. Para tentar solucionar isso, usávamos o chat da ferramenta.”

Quanto à colaboração, o principal desafio citado foi a rejeição do uso da prática pelo par, conforme relato do respondente do projeto 7:

“O principal problema de PPD quanto à colaboração está relacionado com a rejeição de algumas pessoas a prática. Desta forma, acredito também que a PPD tem que ter um forte apelo da alta gerência para evitar isso, bem como o processo de seleção deve escolher pessoas com um perfil colaborativo.”

A falta de uma ferramenta específica para PPD que seja confiável foi relatada por alguns profissionais, o entrevistado do projeto 1 relata:

“Testamos uma série de ferramentas e plugins para a PPD, mas nenhuma era estável o suficiente para ser usada no projeto. Então, optamos pelo Skype, porém acredito que não seja o ideal, acredito que uma ferramenta que seja integrada com uma IDE possa ajudar na produtividade.”

O período de silêncio durante PPD também é um desafio apontado na colaboração apontado pelo desenvolvedor do projeto 2:

“Um dos desafios de PPD que tínhamos é quando o outro desenvolvedor ficava em silêncio durante o pareamento. Esse tipo de postura prejudica muito a prática, então quando identificávamos isso, estimulávamos a conversa. PPD é uma prática que funciona com bastante comunicação”.

Outros desafios citados foram o esforço que a prática cria na colaboração, a qual foi citada na seção anterior, e a distração dos desenvolvedores, conforme relata o respondente do projeto 7:

“Em PPD não temos como saber o que o observador está fazendo, em alguns pareamentos o programador parecia estar disperso. Por isso, acredito que seja muito importante o uso da webcam para tentar manter o foco da PPD.”

A tabela 4.5 abaixo sumariza as categorias identificadas sobre os desafios de PPD:

Tabela 4.5 – Desafios da PPD

Desafios da Programação em Par Distribuída
Esforço
Período de silêncio
Infraestrutura
Ferramenta específica
Idioma
Rejeição da prática
Distração

4.3.7 Comparação com o desenvolvimento sem pareamento

Percebeu-se durante as entrevistas que a comparação da PPD em relação ao desenvolvimento distribuído sem PPD que as respostas foram bem divergentes entre os projetos. Enquanto alguns entrevistados relataram que PPD é uma prática que requer bastante esforço e deve ser usada pontualmente quando necessários, outros apontaram que a PPD é mais efetiva do que o desenvolvimento distribuído sem o pareamento e deve ser mais difundida. Em relação a PP, a maioria dos entrevistados considerou que com

PPD é possível alcançar alguns benefícios da PP, tais como: a qualidade do código, a transferência de conhecimento e a melhoria da comunicação da equipe.

4.3.8 Comentários adicionais

Nas últimas perguntas da entrevista, os respondentes possuíam a possibilidade de acrescentar sugestões para PPD. A participação da maioria dos respondentes foi efetiva, concentrando os comentários em sugestões aplicadas à prática de PPD nos projetos analisados.

Foram citadas sugestões em relação ao desenvolvimento, tais como: a definição de padrões de código entre a equipe para evitar discrepâncias no código entre os pares, a utilização de *Domain-Driven Design* (DDD) para se criar uma linguagem de domínio entre os desenvolvedores e melhorar a colaboração do time. Quanto ao ferramental que apóie PPD citou-se a necessidade de uso de uma ferramenta específica para PPD, com funcionalidades como a gravação das sessões de PPD e um *chat* inserido no próprio ambiente de desenvolvimento. Em relação à infraestrutura uma sugestão foi à utilização de monitores ou TVs que mostrem todo o time durante PPD, dando uma sensação de proximidade física, e a utilização da Webcam durante as sessões de PPD para que seja possível identificar as reações e melhorar a comunicação. Em relação à adoção da prática, recomendou-se: utilizar um projeto piloto nas primeiras sessões para que sejam feitos ajustes e se crie interação entre os desenvolvedores, o uso de sessões curtas ou pausas frequentes para minimizar o desgaste da prática e treinamentos entre a equipe para se criar um conhecimento técnico e de negócio em comum.

4.4 Lições aprendidas

As entrevistas realizadas apresentaram diferentes aspectos da prática de PPD, como ilustrado nas seções anteriores. A seguir destacam-se alguns resultados finais obtidos, apresentados em forma de lições aprendidas, que serão uma das bases para a proposta de um conjunto de boas práticas para PPD.

Lição 1: A PPD requer um esforço adicional à PP e ao desenvolvimento sem pareamento

Os entrevistados relataram que PPD exige menos distração dos desenvolvedores e um esforço maior na colaboração para que a prática seja efetiva. Algumas boas práticas podem ser utilizadas para minimizar esse esforço, tais como: **a utilização de sessões**

curtas com a aplicação de técnicas de gerenciamento de tempo como o Pomodoro¹ que beneficia pausas freqüentes.

Lição 2: A existência de uma ferramenta efetiva e específica para PPD funciona como um facilitador no ambiente da prática

Apesar de apenas um projeto analisado utilizar uma ferramenta específica para PPD, a maioria dos respondentes relatou que a PPD seria mais efetiva se possuísse uma ferramenta de referência, estável e específica para a prática. Os respondentes também citaram que a ferramenta de PPD deveria ter algumas características, como: a integração com um ambiente de desenvolvimento e a gravação de uma sessão de PPD.

Lição 3: PPD necessita de uma infraestrutura apropriada

A infraestrutura foi apontada como um dos fatores de sucesso de PPD pelos respondentes. Uma boa conexão de internet foi a principal característica relatada pelos entrevistados sobre o ambiente de PPD. Aliado a isso, a Webcam foi citada como essencial durante as sessões de PPD, os entrevistados explicaram que sem o vídeo da outra pessoa há o aumento de distração e a perda da sensação da proximidade física. Outra alternativa citada foi o uso de TVs e monitores que mostrem o escritório e toda a equipe.

Lição 4: É importante ter um conhecimento comum sobre o projeto e sobre aspectos técnicos antes das sessões de PPD

Os respondentes relataram que a diferença de conhecimento não é um problema com a PPD, porém informaram que é importante antes das sessões que o time tenha um conhecimento prévio sobre o projeto e sobre a tecnologia empregada. Foram sugeridas, boas práticas como: **reuniões diárias e sessões técnicas** (do tipo *coding dojo*) com vista a melhorar a colaboração.

Lição 5: É importante que se estimule a comunicação durante uma sessão de PPD

A PPD é uma prática que exige uma colaboração mútua entre os pares, para isso é importante evitar o período de silêncio e as distrações. A maioria dos respondentes relatou que apesar da tela ser compartilhada com o par, é importante que quem esteja controlando o teclado **narre e explique** as ações durante as sessões de PPD, isto facilita o acompanhamento e estimula a comunicação.

Lição 6: A utilização de outras estratégias de pareamento é viável com equipes distribuídas

¹ <http://www.pomodorotechnique.com>

Alguns entrevistados relataram que além de utilizarem PPD para desenvolvimento de software, também é utilizado outras formas de pareamento de maneira efetiva. Umas delas é o *Ping Pong Pair Programming* que é uma combinação de PP com a prática de TDD. Outra estratégia utilizada é parear com um analista de teste, mesmo para as tarefas de desenvolvimento, segundo os entrevistados isto ajuda na adequação do código ao negócio.

Lição 7: Promover *feedback* contínuo é uma prática importante para PPD

Os respondentes relatam que o *feedback* após as sessões é muito importante para a constante melhoria do ambiente. Foram citadas como boas práticas: reuniões de retrospectivas e que avaliações após cada sessão de PPD sejam realizadas com o objetivo de verificar a infraestrutura e a colaboração da prática. Um projeto piloto antes da implantação de PPD é uma boa prática que foi relata pelos entrevistados como sendo efetiva na obtenção de *feedback* desde do início do projeto.

Lição 8: PPD pode ser utilizada por equipes auto-organizáveis

A maioria das equipes dos projetos analisados eram auto-organizáveis no que tange a prática de PPD. Elas se organizavam para a criação e o estabelecimento do pares. Apenas um projeto analisado possuía um *coach* responsável neste sentido. Os respondentes acreditam também que o aprendizado da prática possa ser organizado pela própria equipe.

A tabela 4.6 abaixo sintetiza os pontos observados obtidos a partir das lições aprendidas.

Tabela 4.6 – Pontos Observados no Estudo

Lição	Ponto Observado
1	Esforço
2	Ferramental de apoio
3	Infraestrutura
4	Conhecimento (projeto e técnico)
5	Comunicação
6	Estratégias de pareamento
7	<i>Feedback</i> contínuo
8	Equipes auto-organizáveis

4.5 Limitações do estudo de caso

Uma das principais limitações do estudo de caso refere-se ao número de empresas estudadas, restringindo a generalização dos resultados obtidos. Deve-se, entretanto,

destacar que especificamente os resultados, principalmente os da categorização dos elementos, foram sustentados na RSL executada, o que permite um bom grau de segurança nas conclusões obtidas. Isto também é típico do tipo de pesquisa desenvolvida, exploratória e de base qualitativa, possibilitando o uso de inferências nas conclusões obtidas.

5 CONJUNTO DE BOAS PRÁTICAS PARA PDD

Neste capítulo apresenta-se o conjunto de boas práticas para PPD proposto. Na seção 5.1 descreve-se o conjunto de boas práticas e as suas características. A seção 5.2 apresenta as limitações do conjunto proposto.

As características e condições de aplicação da prática de PP e PPD foram obtidas nos principais estudos da literatura a partir da execução da RSL. Adicionalmente, estabeleceu-se um conjunto preliminar de boas práticas para PP e PPD com base nos principais resultados encontrados. Tendo em vista este conjunto preliminar e os múltiplos estudos de casos desenvolvidos, esta pesquisa propõe um conjunto de boas práticas para PPD. Este conjunto tem por objetivo servir de apoio às organizações que utilizam ou pretendem implantar PPD de forma efetiva, explorando os benefícios que esta pode proporcionar. O conjunto de boas práticas foi dimensionado em duas categorias: consequência de utilização da boa prática e o indicador de melhoria.

5.1 Conjunto de boas práticas para PPD

O conjunto de boas práticas para PPD foi criado para atuar como um facilitador nos projetos que utilizam PPD. Além disso, a forma como o conjunto foi concebida permite a identificação e oportunidades de melhoria. A organização do conjunto foi alicerçada em quatro estratégias:

1. A partir dos resultados da RSL, propor um conjunto de práticas preliminar para PP e para PPD, buscando compreender se as práticas específicas para PP são aplicáveis a PPD. Este conjunto preliminar compreende as boas práticas 1, 4, 5 da tabela 5.1;
2. Avaliar se as práticas de PPD identificadas na RSL são corroboradas pelas práticas identificadas a partir dos resultados dos múltiplos estudos de caso executados;
3. Listar as boas práticas obtidas apenas na RSL;
4. Listar as boas práticas obtidas apenas nos múltiplos estudos de caso

A tabela 5.1 apresenta o conjunto de boas práticas proposto nesta pesquisa e a fonte onde foi identificada cada boa prática (a revisão sistemática da literatura – RSL – e os múltiplos estudos de caso – MEC – realizados). A seguir são apresentadas, em detalhes, as boas práticas obtidas a partir das quatro estratégias.

Tabela 5.1 – Conjunto de Boas Práticas

Boa Prática	Consequência	Indicador	Fonte
1. Uso de um <i>guideline</i> (protocolo) para adoção de PPD.	Possibilita que os programadores entendam a prática segundo a política organizacional, observando os seus aspectos de funcionamento como: frequência de troca entre os pares, critérios de formação dos pares, tipo de tarefas em que aconselha o uso de PP, regras do uso da infraestrutura, entre outras coisas.	Ajuda no aprendizado da prática e no estabelecimento de um padrão de código que ajuda na colaboração entre as equipes	RSL - [CAN06]
2. Fazer uma reunião de alinhamento antes da sessão de PPD.	O conflito entre ideias e expectativas entre os pares é um desafio em PPD, uma reunião curta de alinhamento antes minimiza este conflito.	Minimizar os conflitos entre os pares e alinha os objetivos da sessão de PPD	RSL - [ROS10]
3. Realizar treinamentos técnicos para equipe sobre PP e PPD.	Auxilia a equipe a entender melhor os benefícios e desafios da prática.	Minimiza a falta de cumprimento dos papéis de PPD e melhora o conhecimento técnico da equipe sobre a prática.	RSL - [ROS10].
4. Adotar uma Infraestrutura específica.	PP exige comunicação contínua entre os pares, por isso deve ter uma sala apropriada para a prática, com o intuito de não atrapalhar os outros membros da equipe.	Melhoria em variáveis como Comunicação e Colaboração entre os pares.	RSL - [BEV02], [VAN07b], MEC
5. Formação dos pares: profissionais Sênior formando com outro Sênior não é recomendado. Profissional de perfil pleno apenas para tarefas complicadas. Profissionais Junior	Se o objetivo for o aumento da qualidade, recomenda-se formar pares com os profissionais de nível Junior e de nível Pleno. Profissionais de nível Sênior não são recomendados a formarem pares de PP, a não ser que a tarefa seja muito complexa de ser realizada por um programador sênior	Possibilita que os pares possam focar tarefas em que PP tenha mais efetividade , potencializando seus benefícios.	RSL - [Dyb09], MEC

Boa Prática	Consequência	Indicador	Fonte
devem formar pares tanto em tarefas fáceis quanto complicadas.	individualmente.		
6. Possuir um líder de PPD na equipe.	O líder evita impedimentos e impasses entre os pares, bem como tira dúvidas da prática para os iniciantes	Evita conflitos entre personalidades e dúvidas técnicas entre os pares.	RSL- [HAN10], [VAN07b] MEC
7. Uso de uma ferramenta específica para PPD .	Recomenda-se utilizar apenas uma ferramenta que centralize os serviços necessários em PPD, diferentes ferramentas gastam mais tempo durante a prática.	Aperfeiçoa o tempo e evita distrações em PPD.	RSL – [CAN06], [ROS10], MEC
8. Planejar Reuniões Frequentes (<i>brainstorming</i>) para se criar uma visão comum do projeto	Pela distância geográfica o nível de conhecimento do projeto é destoante entre a equipe, o que acaba dificultando o processo de comunicação e colaboração. Com as reuniões os grupos ficam mais integrados sobre o projeto.	Compartilha conhecimento da tarefa, projeto a todos envolvidos em PPD.	RSL - [CAN06], MEC
9. Fornecer Feedback após as sessões de PPD	Com um <i>feedback</i> após cada sessão é possível identificar o que pode ser melhorado no ambiente de PPD.	Constante identificação dos desafios das sessões de PPD.	MEC
10. Utilizar sessões curtas de PPD com pausas frequentes.	A adoção de pausas durante as sessões de PPD possibilita que a prática não se torne exaustiva e mantém o foco na tarefa.	Auxilia a diminuir o esforço da prática, deixando a mais dinâmica .	MEC
11. Planejar um Projeto Piloto antes de adotar a prática	A adoção de um projeto piloto colabora para o ajuste de um ambiente de PPD. Além de tornar a prática mais familiar entre os desenvolvedores.	Identifica os desafios e aprimora o ambiente necessário para a adoção de PPD.	MEC
12. O driver deve narrar as ações durante as sessões de PPD	Evita que as sessões de PPD tenham períodos de silêncio ou distrações.	Melhora a comunicação entre os pares e o acompanhamento do <i>observer</i> .	MEC

A boa prática 1 diz a respeito ao uso de um *guideline* para adoção de PPD. Segundo Canfora, a utilização de um *guideline* para PPD (boa prática 1) que possuísse os aspectos de funcionamento como: frequência de troca entre os pares, critérios de formação dos pares, tipo de tarefas em que se recomenda o uso de PP, regras do uso da infraestrutura; facilitaria o entendimento da aplicação da prática pela equipe [CAN06].

De acordo com Rosen [ROS10], o conflito entre objetivos entre os pares é um grande desafio na utilização da PPD. Para solucionar isto reuniões curtas frequentes podem ser realizadas para que possam ser alinhados quais são os objetivos a serem tratados na sessão (boa prática 2). A reunião de alinhamento evita as discussões durante a sessão que não agregam ao objetivo da tarefa. Outro desafio apontado pelo autor é falta de cumprimento dos papéis de PPD (*driver* e *observer*), uma das soluções encontradas é a realização de treinamento sobre a prática de PP e PPD (boa prática 3), para que a equipe (principalmente aqueles que nunca tiveram experiência) possam ter mais conhecimento técnico sobre a prática.

A boa prática 4 se refere à necessidade de ser ter uma infraestrutura específica para PPD, Bevan relata a importância de ser ter uma sala específica, isto possibilitaria não atrapalhar outros membros da equipe que não utilizam a prática, além de reunir mais equipamentos necessários como teclados ou monitores extras [BEV02]. Alguns entrevistados relataram fazer uso de TVs grandes e monitores que também auxiliam a criar uma sensação de proximidade física entre os pares.

A boa prática 5 está relacionada com a formação dos pares, de acordo com Dyba a diferença de experiência entre os desenvolvedores gera diferentes tipos de resultados [Dyb09]. O autor relata que o uso de dois profissionais Sênior não é recomendado, apenas em casos onde a tarefa possui um nível crítico bem elevado. Os profissionais menos experientes (Junior) devem sempre fazer o pareamento visando garantir o compartilhamento do conhecimento. A variável de qualidade de código tem um efeito positivo quando se combina profissionais Pleno com profissionais Junior.

Alguns projetos analisados nos múltiplos estudos de caso apontaram que o critério de formação dos pares foi a experiência dos desenvolvedores. Quando um profissional Junior, isto é, com pouca experiência entrava na equipe, já participava do pareamento. Apesar da boa prática encontrada na RSL ser mais abrangente quanto ao tipo de experiência, os resultados na indústria apontaram que apenas os profissionais com

menos experiência tinham uma atenção maior, sendo recomendado que eles participassem do pareamento tanto em tarefas complicadas quanto em triviais com o objetivo de adquirir conhecimento técnico.

A boa prática 6 se refere à presença de um líder (*coach*) de PP na equipe, segundo Hannay essa boa prática ajuda a evitar impasses entre os pares e possibilita que as dúvidas referentes a prática de PP possam sem rapidamente respondidas [HAN10]. A boa prática 1 vai ao encontro da minimização do efeito negativo da diferença de personalidades, apresentado na RSL (capítulo 2). A presença de um líder de PPD que atuasse no papel de *coach* e facilitador foi apontada por um projeto dos múltiplos estudos de caso como sendo uma prática que ajuda na resolução de dúvidas técnicas e conflitos. Além disso, o *coach* foi responsável pela formação dos pares na PPD.

A boa prática 7 se refere ao uso de uma ferramenta específica para PPD (boa prática 6). Segundo Canfora [CAN06], um dos principais problemas na PPD é a falta de uma apropriada plataforma ferramental. Ele ainda sugere que o ferramental específico seja integrado com um sistema de gerência de configuração. Rosen relata que uma ferramenta específica para PPD facilita o uso de um único ambiente para o desenvolvedor, sem que seja necessária a alternância entre as janelas, aperfeiçoando a agilidade da prática [ROS10]. O autor ainda afirma que um ferramental específico que compartilhe a tela e vídeo auxilia na diminuição da distração entre os desenvolvedores [ROS10]. Alguns entrevistados sugeriram algumas funcionalidades para uma ferramenta específica de PPD, tais como: a gravação de sessões de PPD e um *chat* integrado ao ambiente de desenvolvimento. Ainda segundo os resultados das entrevistas, a prática de PPD carece de uma ferramenta que seja referência, consolidada na indústria.

A boa prática 8 diz a respeito do planejamento de reuniões frequentes, segundo, Canfora a distribuição dos pares tende a diminuir as discussões, como consequência, há a falta de uma visão única do projeto [CAN06]. Segundo ele, reuniões com a equipe podem ser planejadas durante todas as fases do projeto, principalmente no início. Os entrevistados também ratificaram a adoção desta boa prática, relatando aplicar reuniões técnicas e de projeto antes das sessões de PPD, além de reuniões diárias estabelecidas por equipe que utilizam o método *Scrum*.

O *feedback* após o término de cada sessão (boa prática 9) foi relatado como uma importante fonte de identificação dos desafios e melhoria do ambiente de PPD. De acordo

com os entrevistados, é importante fazer algumas perguntas básicas para estimular o *feedback*, tais como:

1. “Qual sua avaliação da sessão de PPD?”
2. “Quais os pontos podem ser melhorados no ambiente de PPD?”
3. “Quais estratégias podem ser adotadas para melhorar a sessão de PPD?”

A maioria dos respondentes relatou que a PPD é uma prática exaustiva, exigindo mais esforço da equipe. Para minimizar isto, uma boa prática indicada foi o uso de sessões curtas com pausas freqüentes (boa prática 10). A consequência da utilização desta boa prática é a diminuição do esforço, ajudando a manter o foco na tarefa. Uma estratégia citada foi a utilização de técnicas de gerenciamento de tempo, como a técnica Pomodoro¹.

Antes da adoção de PPD é importante compreender as especificidades do ambiente, desta forma alguns entrevistados relataram que seria importante planejar e executar um projeto piloto de PPD. O projeto piloto tem o papel (boa prática 11) de identificar os desafios e planejar as estratégias para melhorar o ambiente de PPD, além de ajudar os colaboradores a se familiarizar com a prática.

Outra boa prática relatada pelos entrevistados foi em relação ao papel do *driver*. Segundo eles, durante a sessão de PPD o *driver* deve constantemente narrar as suas ações no código (boa prática 12). A adoção desta boa prática implica na diminuição dos períodos de silêncio que são prejudiciais a PPD como apontado na RSL. Outro benefício é o acompanhamento do *observer*, o qual também é estimulado a buscar a comunicação durante a sessão.

5.2 Limitações do conjunto de boas práticas proposto

O conjunto de boas práticas para PPD proposto trata-se de uma proposta inicial, portanto não foi possível se aprofundar na análise empírica da adoção das boas práticas propostas. O conjunto não apresenta características específicas em relação ao tamanho da equipe ou ao processo de software utilizado pela organização, tais questões ainda necessitam de análise.

¹ <http://www.pomodorotechnique.com>

6 CONCLUSÃO

Com o avanço da adoção do DDS nas organizações, pesquisadores da área de ES tem se defrontado com desafios e com a necessidade de propor novas soluções [GOM12]. Paasivara [PAA09] relata que as práticas dos métodos ágeis são uma estratégia para minimizar parte dos desafios das equipes de DDS. Neste contexto, a PPD é uma prática ágil do método XP, que com sua crescente adoção, necessita de mais investigação [BAN10].

O conjunto de boas práticas para PPD proposto é uma tentativa de contribuir na adoção de práticas ágeis no contexto do DDS. Os resultados encontrados apresentam evidências de que a área de ES necessita de mais pesquisas voltadas para estes novos desafios que o ambiente de DDS está trazendo. Do ponto científico, a legitimação do conjunto de boa prática é decorrente do processo pesquisa como um todo, representado no desenho de pesquisa e desdoblado nas diversas fases deste estudo, conforme apresentado no capítulo 3.

6.1 Contribuições da pesquisa

A principal contribuição desta pesquisa é a proposta de um conjunto de boas práticas para PPD (apresentado no capítulo 5), que contempla evidências da literatura e a experiência de profissionais da indústria. O conjunto de boas práticas soma-se resultados empíricos encontrados na RSL, como mais uma contribuição para responder aos desafios da área de DDS, especificamente da prática de PPD. O objetivo desta pesquisa é que o conjunto proposto contribua como um facilitador para empresas que utilizam ou desejam utilizar PPD.

A questão de pesquisa desta pesquisa foi respondida ao longo do processo de formulação do conjunto de boas práticas proposto, descreveram-se o estado da arte de PP e PPD, apresentado a RSL executada (capítulo 2), nesta identificaram os principais benefícios e desafios. Adicionalmente, os resultados da RSL formaram um conjunto preliminar de boas práticas para PP e PPD. Este conjunto foi ampliado e refinado com as experiências dos profissionais, obtidas nos múltiplos estudos de caso (capítulo 4).

6.2 Trabalhos futuros

Identifica-se grande potencial de crescimento nesta linha de pesquisa, onde os pontos fortes envolvem uma parceria estável entre a academia e a indústria, criando condições de experimentação e aprendizagem únicas, decorrentes de uma sinergia positiva entre os parceiros.

Como pesquisas futuras, sugere-se:

- Continuidade da avaliação empírica de PPD por meio do ciclo de Mafra [MAF06], adaptado de [SHU01], conforme figura 6.1. Este ciclo tem por finalidade o desenvolvimento de propostas empíricas na área de ES que sejam transferidas para indústria. Parte deste ciclo que inclui o estudo de viabilidade e o estudo de observação foram realizados pelos múltiplos estudos de caso.

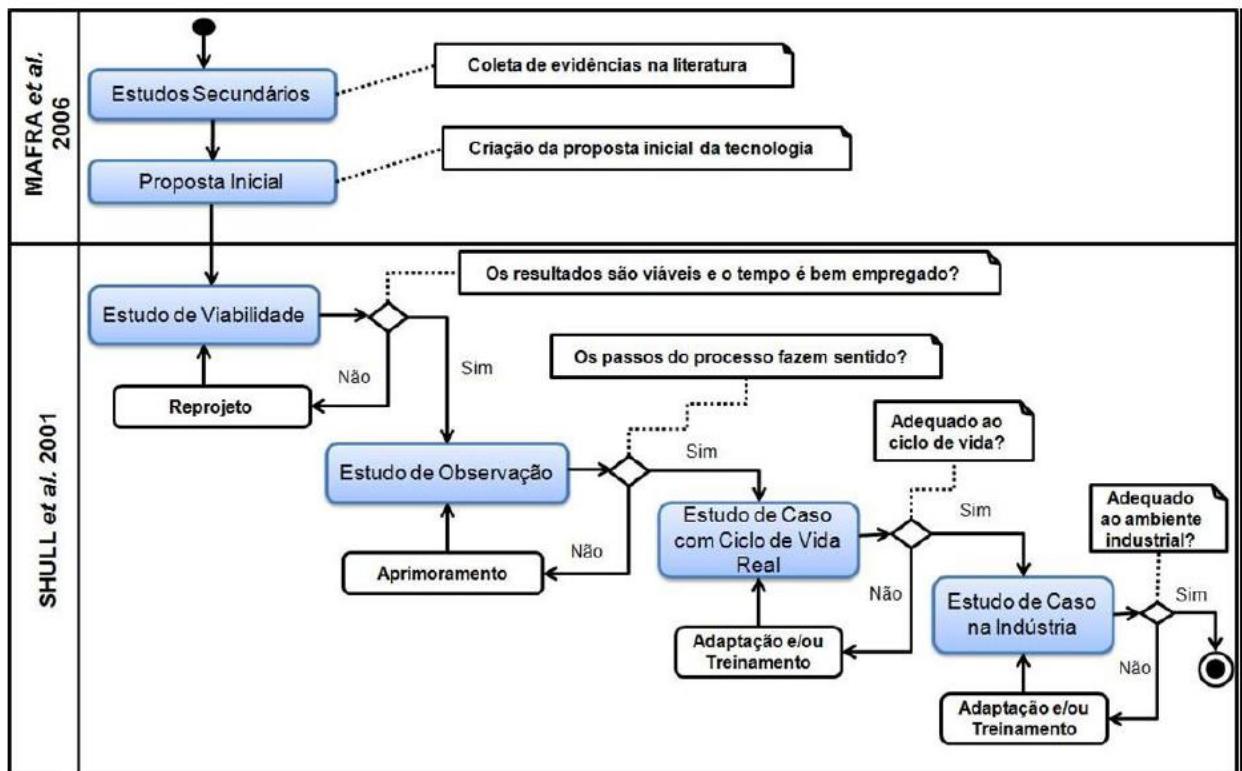


Figura 6.1 – Ciclo de Mafra [MAF06]

- Aprofundar os estudos sobre as boas práticas identificadas apenas na RSL e analisar empiricamente se as evidências que elas sugerem podem ser úteis às organizações.
- Aprofundar os estudos sobre as boas práticas identificadas apenas nos múltiplos estudos de caso, provendo mais evidências empíricas para que estas tenham uma contribuição mais forte para as organizações.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ARI07] Arisholm, E.; Gallis, H.; Dyba, T.; Sjoberg, D. "Evaluating pair programming with respect to system complexity and programmer expertise". *IEEE Transactions on Software Engineering*, vol. 33-2, Fev-2007, pp. 65–86.
- [AUD07] Audy, J. L. N.; Prikladnicki, R. "Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas". Rio de Janeiro: Elsevier, 2007, 211p.
- [BAH02] Baheti, P.; Gehringer, E.; Stotts, D. "Exploring the efficacy of distributed pair programming". In: Extreme Programming and Agile Methods—XP Agile Universe, 2002, pp.387-410.
- [BAN10] Bandukda, M.; Nasir, Z. "Efficacy of distributed pair programming," In: International Conference on Information and Emerging Technologies, 2010, pp.1-6.
- [BAS08] Bassi, D. "Experiências com Desenvolvimento Ágil". Dissertação de Mestrado, Programação de Pós-Graduação em Ciência da Computação, IME-USP, 2008, 170p.
- [BEC01] Beck, K.; et al. "Manifesto for Agile Software Development". Capturado em: www.agilemanifesto.org, Janeiro 2013.
- [BEC04] Beck, K.; Andres, C. "Extreme Programming Explained: Embrace Change". Addison-Wesley Professional, 2004, 2 .ed, 224p.
- [BEG08] Begel, A.; Nagappan, N. "Pair programming: what's in it for me?". In: International symposium on Empirical software engineering and measurement, 2008, pp. 120-128.
- [BEV02] Bevan, J.; Werner, L.; McDowell, C. "Guidelines for the use of pair programming in a freshman programming class", In: Conference on Software Engineering Education and Training, 2002, pp.100-107.
- [BIP07] Bipp, T; Lepper, A; Schmedding, D. "Pair programming in software development teams - an empirical study of its benefits". *Information and Software Technology*, vol. 50-3, Jun-2007, pp. 231–240.
- [BRA08] Braught, G.; Eby, L.; Wahls, T. "The effects of pair-programming on individual programming skill". In: SIGCSE, 2008, pp. 200-204.

- [BRA11] Braught, G.; Wahls, T.; Eby, L. "The case for pair programming in the computer science classroom". *ACM Transactions on Computing Education*, Fev-2011, vol.11-1, 2011.
- [BRE09] Brereton, P.; Turner, M.; Kaur, R. "Pair programming as a teaching tool: a student review of empirical studies". In: Software Engineering Education and Training, 2009, pp. 240-247.
- [CAN03] Canfora, G.; Cimitle, A.; Visaggio, C.. "Lessons learned about distributed pair programming: what the knowledge needs to address?". In: Proceedings of the Twelfth IEEE International Conference of Enabling Technologies; Infrastructure for Collaborative Enterprises, 2003, pp. 314 - 319 .
- [CAN06] Canfora,G.; Cimitle,A.; Visaggio,C.;Di Lucca,G. "How distribution affects the success of pair programming". In:International Journal of Software Engineering and Knowledge Engineering, 2006, pp. 293 - 313.
- [CAO05] Cao, L.; Xu, L. "Activity Patterns of Pair Programming," In: Hawaii International Conference on System Sciences , 2005, 6p.
- [CAR99] Carmel, E. "Global Software Teams – Collaborating Across Borders and Time-Zones". Prentice Hall, 1999, 296p.
- [CAR07] Carver, C.; Henderson, L.; Lulu, He.; Hodges, J.; Reese, D.; , "Increased Retention of Early Computer Science and Software Engineering Students Using Pair Programming,". In: Software Engineering Education & Training, 2007, pp.115-122.
- [CHO07a] Choi ,K; Deek, F; Im, I. "Exploring the underlying aspects of pair programming: The impact of personality". *Information and Software Technology*, vol. 50-11),Nov-2007, pp. 1114–1126.
- [CHO07b] Chong, J; Hurlbutt, T. "The Social Dynamics of Pair Programming". In: International Conference on Software Engenieerig , 2007, pp.354-363.
- [CHI08] Chigona, W.; Pollock, M. "Pair programming for information systems students new to programming: Students' experiences and teachers' challenges," In: Portland International Conference on Management of Engineering & Technology, 2008, pp.1587-1594.
- [COC04] Cockburn, A. "Crystal Clear, A Human-Powered Methodology for Small Teams". Addison-Wesley Professional, 2004, 336p..
- [COC06] Cockburn, A. "Agile Software Development: The Cooperative Game". Addison-Wesley Professional, 2006, 504p.
- [DYB07] Dyba, T.; Arisholm, E.; Sjoberg, D.I.K.; Hannay, J.E.; Shull, F.; , "Are Two Heads Better than One? On the Effectiveness of Pair Programming.". In: *IEEE Software*, vol. 24-6, Nov.-Dec. 2007, pp.12-15.

- [DYB08] Dyba, T.; Dingsøyr, T. "Empirical studies of agile software development: A systematic review". *Information Software Technology Journal*, vol. 50-10, Ago-2008, pp. 833-859.
- [DYB09] Dyba, T.; Arisholm, E.; Sjoberg, D.I.K.; Hannay, J.E.; Shull, F." The effectiveness of pair programming: A meta-analysis". *Inf. Softw. Technol.*, vol. 51-7, Jul-2009, pp. 1110-1122.
- [FRE05] Freitas, V. "APSEE-Global: a Model of Processes Management of Distributed Software Processes". Dissertação de Mestrado, PPGC, UFRGS, 2005, 155p.
- [FRO11] Fronza, I.; Sillitti, A. ; Succi, G.; Vlasenko, J. "Analysing the usage of tools in pair programming sessions". In: XP, 2011, 11p.
- [GIR12] Giri, M.; Dewangan, M. "A Study of Pair Programming in the Context of Facilitating the Team Building". In: Second International Conference on Advanced Computing & Communication Technologies (ACCT), 2012, vol., no., pp.20-23, 7-8 Jan. 2012.
- [GOM12] Gomes, V.; Marczak, S. "Problems? We All Know We Have Them. Do We Have Solutions Too? A Literature Review on Problems and Their Solutions in Global Software Development". In: International Conference on Global Software Development, 2012, pp.154-158, 2012.
- [HAO11] Hao, J. "Distributed Pair Programming in Global Software Development". Dissertação de Mestrado, School of Informatics, Universidade de Edimburgo, 2011, 94p.
- [HAN05] Hanks, B."Student performance in CS1 with distributed pair programming". In: SIGCSE Bull, 2005, pp. 316-320.
- [HAN08] Hanks, B. "Problems encountered by novice pair programmers". *ACM Journal on Educational Resources in Computing*, vol. 7-4, Jan-2008.
- [HAN10] Hannay, J.; Arisholm, E.; Engvik, H.; Sjoberg, D." Effects of personality on pair programming". In: IEEE Transactions on Software Engineering, vol. 36-1, Fev-2010, pp.61-80.
- [HAN11] Han, L; Wenjuan X. "An experimental research of the pair programming in java programming course," In: International Conference on e-Education, Entertainment and e-Management (ICEEE), 2011, pp.257-260.
- [HO04] Ho, C.; Raha, C.; Gehringer,E.;Williams, L. "Sangam: a distributed pair programming plug-in for eclipse". In: OOPSLA, 2004, pp. 73-77.
- [HUL05] Hulkko,H; Abrahamsson , P." A multiple case study on the impact of pair programming on product quality". In: Proceedings of the 27th international conference on Software engineering (ICSE '), 2005, pp. 495-504.

- [HER01] Herbsleb, J.; Moitra, D. "Global Software Development". *IEEE Software*, California, vol. 16-2, Mar- Abr 2001, pp. 16-20.
- [KIT07] Kitchenham, B; Charters, S. "Guidelines for performing Systematic Literature Reviews in Software Engineering", Technical Report, Keele University and Durham University, 2007, 65p.
- [LUI06] Lui, K.; Chan, K. "Pair programming productivity: Novice-novice vs. expert-expert". *International Journal of Human Computer Studies*, vol. 64-9, Set-2006, pp.915–925.
- [MAF06] Mafra, S. N.; Barcelos, R. F.; Travassos, G. H. "Aplicando uma Metodologia Baseada em Evidencia na Definição de Novas Tecnologias de Software". In: Simposio Brasileiro de Engenharia de Software, 2006, Florianopolis, 2006, pp. 239-254.
- [MCD02] McDowell, C.; Werner, L. ; Bulock,H. ; Fernald, J. "The effects of pair-programming on performance in an introductory programming course". In: SIGCSE technical symposium on Computer science education, 2002, pp. 38–42.
- [MEN05] Mendes,E.; Basil, L.; Fakhri,A.; Reilly, A." Investigating pair-programming in a 2-year software development and design computer science course".In: SIGCSE Bull., 2005, pp. 296–300.
- [MOZ10] Mozzato, A. R.; Grzybovski, D. "Análise de Conteúdo como Técnica de Análise de Dados Qualitativos no Campo da Administração: Potencial e Desafios", *Revista de Administração Contemporânea*, vol. 15-4, Jul-Ago 2010, pp. 731-747.
- [MUL04] Muller, M.M.; Padberg, F. "An empirical study about the feelgood factor in pair programming". In: Proceedings. of 10th International Symposium on Software Metrics, 2004,pp. 151- 158..
- [MUL07] Müller,M. "Do programmer pairs make different mistakes than solo programmers?". *J. Syst. Softw.*, vol.80-9, Set-2007, pp.1460–1471.
- [NAG03] Nagappan, N.; Williams, L., Ferzli, M.; Wiebe, E.; Miller, K.; Balik, S. "Improving the CS1 experience with pair programming". In: SIGCSE Bull, 2003, pp. 359-362.
- [PAA09] Paasivaara, M.; Durasiewicz, S.; Lassenius, C. In: "Using Scrum in Distributed Agile Development: A Multiple Case Study". In: ICGSE 2009, pp. 195-204.
- [PRI03] Prikladnicki, R. "MuNDDoS - Um modelo de referência para Desenvolvimento Distribuído de Software". Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, PUCRS, 2003, 144p.
- [PRE11] Pressman, R. "Software Engineering: A Practitioner's Approach". Macgraw-hill, 7ed, 2011, 880p.

- [POP03] Poppendieck, M.; Poppendieck, T. "Lean Software Development: An Agile Toolkit". Adisson-Wesley Professional, 2003, 240p.
- [RAM08] Ramli, N.; Fauzi S. "The effects of pair programming in programming language subject". In: International Symposium on Information Technology, 2008, pp.1-4.
- [ROS10] Rosen, E.; Salinger, S.; Oezbek, C. "Project Kick-off with Distributed Pair Programming". In: Workshop of Psychology of Programming Interest Group, Madrid, 2010, 15p.
- [SAL09] Salleh, N.; Mendes, E.; Grundy, J.; Burch, G. 2009. "An empirical study of the effects of personality in pair programming using the five-factor model". In: International Symposium on Empirical Software Engineering and Measurement, 2009, 12p.
- [SAL11] Salleh, N.; Mendes, E.; Grundy, J. "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review". *IEEE Transactions on Software Engineering*, vol.37-4, Jul-Ago. 2011, pp. 509–525.
- [SIM08] Simon, B;Hanks, B. "First-year students' impressions of pair programming in CS1". *J. Educ. Resour. Comput.*, vol. 7-4, January 2008.
- [SIS09] Sison, R. "Investigating the Effect of Pair Programming and Software Size on Software Quality and Programmer Productivity". In: Software Engineering Conference, 2009, pp.187-193.
- [SMI12] Smite, D. ; Wohlin, C .; Galvina, Z. ; Prikladnicki, R. "An Empirically Based Terminology and Taxonomy for Global Software". *Engineering. Empirical Software Engineering: An International Journal*, Jul 2012.
- [SOM11] Sommerville, I. "Software Engineering". São Paulo: Pearson Addison-Wesley, 9 ed., 2011.
- [SCH95] Schwaber, K. "The Scrum Development Process". In: Conference on object oriented programming systems, languages and applications (OOPSLA), 1995, pp. 117-134.
- [SHU01] Shull F., Carver, J.; Travassos, G. In: "An empirical methodology for introducing software processes ". In: ESEC / SIGSOFT FSE, pp. 288-296, 2001.
- [VAN05] Vanhanen, J.; Lassenius, C. "Effects of pair programming at the development team level: an experiment". In: Empirical Software Engineering, 2005, pp. 17-18.
- [VAN07a] Vanhanen, J.; Lassenius, C. "Perceived Effects of Pair Programming in an Industrial Context". In: EUROMICRO Conference on Software Engineering and Advanced Applications, 2007, pp.211-218.

- [VAN07b] Vanhanen, J.; Lassenius, C.; Mantyla, M.V. "Issues and Tactics when Adopting Pair Programming: A Longitudinal Case Study". In: Software Engineering Advances, 2007, pp. 25-31.
- [WAL09] Walle, T.; Hannay, J.E. "Personality and the nature of collaboration in pair programming". In: Empirical Software Engineering and Measurement, 2009, pp.203-213.
- [WIE03] Wiebe, E; Williams, L; Petlick, J; Balik, S; Miller, C; Ferzli, M. "Pair programming in introductory programming labs" In: American Society for Engineering Education Annual Conference & Exposition, 2003, 12p.
- [WIE06] Wieringa, R; Maiden, N; Rolland, C. "Requirements. Engineering Paper Classification and Evaluation Criteria: a Proposal and a Discussion". *Journal of Requir. Eng.*, vol. 11-1, Nov - 2005, pp. 102-107.
- [WIL03] Williams, L.; McDowell, C.; Nagappan, N.; Fernald, J.; Werner, L.; , "Building pair programming knowledge through a family of experiments.". In: Proceedings of International Symposium on Empirical Software Engineering, 2003,pp. 143- 152.
- [WIL07] Williams, L. "Lessons learned from seven years of pair programming at north carolina state university". In:SIGCSE Bull., 2007, pp. 79–83.
- [WIL08] Williams, L.; McCrickard, D.S.; Layman, L.; Hussein, K."Eleven Guidelines for Implementing Pair Programming in the Classroom". In: AGILE '08. Conference 2008, pp.445-452
- [YIN10] Yin, R. K. "Estudo de Caso Planejamento e Métodos". Porto Alegre: Bookman, 2010, 248p.
- [ZAC11] Zacharis,N. "Measuring the effects of virtual pair programming in an introductory programming java course".IEEE Transactions on Education, vol. 54-, Fev-2011, pp. 168–170.

APÊNDICE A

Protocolo de Revisão Sistemática da Literatura

Bernardo José da Silva Estácio

1. Formulação da Questão

1.1 Questões de Pesquisa

O trabalho tem a pretensão de responder as seguintes questões:

- QP1: O que se sabe sobre a utilização da programação em par e a utilização da programação em par distribuída?
- QP2: Em que condições a programação em par funciona?
- QP3: Em que condições a programação em par distribuída funciona?

1.2 Qualidade e Amplitude da Questão

1.2.1 **Problema:** Os métodos ágeis de desenvolvimento têm ganhado o interesse do mercado quanto à adoção de suas práticas e atividades [Dyb08]. Contudo, os efeitos das práticas no desenvolvimento de software ainda são desconhecidos ou escassos do ponto de vista científico. Uma dessas práticas, a Programação em Par, estabelecida no método XP, influí diretamente na qualidade do produto [Mcd02]. Desta forma, faz-se necessário saber quais os resultados de investigações científicas já existem em relação à Programação em Par, sendo possível verificar como é feita sua adoção no contexto de mercado e educacional.

1.2.2 **Palavras Chaves – Sinônimos:**

1. Pair Programming
2. eXtreme Programming
3. XP
4. Agile Software Development

1.2.3 **Intervenção:** Neste caso, busca-se analisar a programação em par, os termos de intervenção são:

1. Pair Programming
2. Pair-programming

- 1.2.4 **Efeito:** Pretende-se identificar evidências empíricas da prática de Programação em Par no mercado e levantar novas iniciativas de pesquisas no assunto.
- 1.2.5 **Outcome:** Os resultados serão medidos por meio dos benefícios encontrados pela programação em par, estes são:
- Limitations
 - Best practice
 - Benefits
 - Advantages
 - Disadvantages
 - Design
 - Strategies
- 1.2.6 **População:** Pela intervenção definida, a população alvo a ser pesquisada são os projetos que utilizam desenvolvimento ágil de software, desta forma os termos a ser listados são:
1. Agile Software Project
 2. Agile Software Development
 3. Agile Software
 4. eXtreme Programming
 5. XP
- 1.2.7 **Aplicação:** Esta revisão sistemática se aplica para pesquisadores da Academia que estejam desenvolvendo pesquisas científicas e necessitem de informações inerentes. Também pode vim a servir para profissionais de Mercado que possuem o interesse em informações importantes a respeito do assunto.
- 1.2.8 **Experimental Design:** Não será utilizado nenhum método estatístico.
- 1.2.9 **Artigos de Controle**
- Baheti, P. "Assessing distributed pair programming". In Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, OOPSLA '02, pages 50–51, New York, NY, USA, 2002. ACM.

- Canfora, G; Cimitile, A; . Di Lucca,G; Visaggios, C. "How distribution affects the success of pair programming".In: International Journal of Software Engineering and Knowledge Engineering, 16(2):293–313, 2006.
- Dyba, T.; Arisholm, E.; Sjoberg, D.I.K.; Hannay, J.E.; Shull, F.". The effectiveness of pair programming: A meta-analysis".In: Inf. Softw. Technol. 51, 7 (July 2009), 1110-1122.".
- McDowell, C; Werner, L; Bullock, H; Fernald, J. "The impact of pair programming on student performance, perception and persistence". In: Proceedings of the 25th International Conference on Software Engineering (ICSE '03). IEEE Computer Society,2003, Washington, DC, USA, 602-607.
- Salleh, N.; Mendes, E.; Grundy, J. "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review". In:. IEEE Transactions on Software Engineering, 37(4):509–525, 2011.

2. Seleção de Fontes

2.1. Definição dos Critérios para Seleção de Fontes

- Artigos de conferência e periódicos;
- Base de dados atualizada;
- Estudos empíricos ou relatos de experiência contidos de forma clara;

2.2. Idioma dos Estudos

O idioma utilizado é o inglês pela quantidade de publicações em conferências e periódicos neste idioma.

2.3. Identificação das Fontes

- **Método de Busca das Fontes:** Pesquisas pela engines de pesquisa científica.

- **String de Busca:**

1) String PICCO

População: projetos de desenvolvimento ágil de software com o método XP
 $P := (\text{ Agile Software Project } \text{ or } \text{ Agile Software Development } \text{ or } \text{ or } \text{ eXtremme Programming } \text{ or } \text{ XP})$

Intervenção: Prática da programação em par

I := (Pair programming programming practice pair
programming practice)

Outcome O := (Limitations <or> Best Practices <or> Benefits

<or>Advantages <or> Disadvantages <or> Design<or> Strategies)

String de busca final: P <and> I <and> O

2) String baseada em revisões sistemáticas anteriores

“Pair Programming” OR “Pair-Programming” [SAL11]

- **Lista de Fontes:**

IEEExplore

ACM Digital Library

Scopus

ISI Web of Knowledge

Compedex

Willey

3. Seleção de Estudos

3.1 Critérios de Inclusão

- Artigos de natureza qualitativa e/ou quantitativa que relatam a prática de Programação em Par em projetos que utilizem métodos ágeis.
- Artigos publicados a partir de 2001, data de publicação do Manifesto Ágil.

3.2 Critérios de Exclusão

Os artigos serão excluídos da pesquisa, caso tiverem relação com estes critérios:

- Artigos que não envolvam processo de desenvolvimento Software e Engenharia de Software.
- Artigos que não tratam de Métodos Ágeis de Desenvolvimento de Software.
- Artigos que não abordam a prática de Programação em Par.
- Estudos que não sejam em sua totalidade no idioma inglês.
- *Short papers.*

3.3 Procedimentos de Seleção de estudos

A seleção dos trabalhos utilizados na revisão se deu por meio de três etapas:

1. Identificação dos artigos obtidos nas *engines* de busca.
2. Exclusão dos artigos baseada na leitura do título, abstract e palavras chave.
3. Exclusão dos artigos baseada na leitura da Introdução e Considerações Finais.
4. Exclusão dos artigos baseada no resultado do *checklist* de avaliação de qualidade.
5. Leitura por completo e análise critica dos artigos.

3.4 Referências Bibliográficas

Dyba, T.; Dingsøyr, T. "Empirical studies of agile software development: A systematic review". *Information Software Technology Journal*, vol. 50-10, Ago-2008, pp. 833-859.

Mcdowell, C.; Werner, L. ;Bulock,H. ; Fernald, J. "The effects of pair-programming on performance in an introductory programming course". In: SIGCSE technical symposium on Computer science education, 2002, pp. 38–42.

APÊNDICE B

Referências dos Estudos Selecionados na RSL

E01	Arisholm, E.; Gallis, H.; Dyba, T.; Sjoberg, D. "Evaluating pair programming with respect to system complexity and programmer expertise". <i>IEEE Transactions on Software Engineering</i> , vol. 33-2, Fev-2007, pp. 65–86.
E02	Baheti, P. "Assessing distributed pair programming". In: OOPSLA, 2002, pp. 50–51.
E03	Bandukda, M.; Nasir, Z. "Efficacy of distributed pair programming," In: International Conference on Information and Emerging Technologies, 2010, pp.1-6.
E04	Begel, A.; Nagappan., N. "Pair programming: what's in it for me?". In: International symposium on Empirical software engineering and measurement, 2008, pp. 120-128.
E05	Bevan, J.; Werner, L.; McDowell, C. "Guidelines for the use of pair programming in a freshman programming class", In: Conference on Software Engineering Education and Training, 2002, pp.100-107.
E06	Bipp, T; Lepper, A; Schmedding, D. "Pair programming in software development teams - an empirical study of its benefits". <i>Information and Software Technology</i> , vol. 50-3, Jun-2007, pp. 231–240.
E07	Boyer, K.; Dwight, A.; Fondren, A.; Vouk, M.; Lester, J. "A development environment for distributed synchronous collaborative programming". In: SIGCSE, 2008, pp. 158-162.
E08	Braught, G.; Wahls, T.; Eby, L. "The case for pair programming in the computer science classroom". <i>ACM Transactions on Computing Education</i> , Fev-2011, vol.11-1, 2011.
E09	Braught, G.; Eby, L.;Wahls, T. "The effects of pair-programming on individual programming skill". In: SIGCSE, 2008, pp. 200-204.
E10	Brereton, P.; Turner, M.; Kaur, R. "Pair programming as a teaching tool: a student review of empirical studies". In: Software Engineering Education and Training, 2009, pp. 240-247.
E11	Bryant, S.; Romero, P.; du Boulay, B. "Pair programming and the mysterious role of the navigator". <i>International Journal of Human Computer Studies</i> , vol. 66-7, Jul-2008, pp. 519–529.
E12	Canfora, G.; Cimitle, A.; Visaggio, C.. "Lessons learned about distributed pair programming: what the knowledge needs to address?". In: Proceedings of the Twelfth IEEE International Conference of Enabling Technologies; Infrastructure for Collaborative Enterprises, 2003, pp. 314 – 319.
E13	Canfora, G.; Cimitle, A.; Visaggio,C.; Di Lucca,G. "How distribution affects the success of pair programming". In:International Journal of Software Engineering and Knowledge Engineering, 2006, pp. 293-313.
E14	Cao, L.; Xu, L. "Activity Patterns of Pair Programming," In: Hawaii International Conference on System Sciences , 2005, 6p.
E15	Carver, C.; Henderson, L.; Lulu, He.; Hodges, J.; Reese, D.; , "Increased Retention of Early Computer Science and Software Engineering Students Using Pair Programming, ". In: Software Engineering Education & Training, 2007, pp.115-122.
E16	Chigona, W.; Pollock, M. "Pair programming for information systems students new to programming: Students' experiences and teachers' challenges," In: Portland International Conference on Management of Engineering & Technology, 2008, pp.1587-1594.
E17	Choi ,K; Deek, F; Im, I. "Exploring the underlying aspects of pair programming: The impact of personality". <i>Information and Software Technology</i> , vol. 50-11),Nov-2007,

	pp. 1114–1126.
E18	Chong, J; Hurlbutt, T. "The Social Dynamics of Pair Programming". In: International Conference on Software Engineering , 2007, pp.354-363.
E19	Dou, W.; He, W. "Compatibility and Requirements Analysis of Distributed Pair Programming". In: International Workshop on Education Technology and Computer Science, 2010, pp.467-470.
E20	Dou, W.; He, W. "A Preliminary Design of Distributed Pair Programming System," In: Second International Workshop on Education Technology and Computer Science (ETCS), 2010, pp.256-259.
E21	Dou, W.; Hong, K.; He, W. "A conversation model of collaborative pair programming based on language/action theory". In: International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2010 14th, pp.7-12.
E22	Dou, W.; Hong, K.; Zhang, X. "A Framework of Distributed Pair Programming System,". In: International Conference on Computational Intelligence and Software Engineering, 2009, pp.1-4.
E23	Dyba, T.; Arisholm, E.; Sjoberg, D.I.K.; Hannay, J.E.; Shull, F.; , "Are Two Heads Better than One? On the Effectiveness of Pair Programming". In: <i>IEEE Software</i> , vol. 24-6, Nov.-Dec. 2007, pp.12-15.
E24	Dyba, T.; Arisholm, E.; Sjoberg, D.I.K.; Hannay, J.E.; Shull, F." The effectiveness of pair programming: A meta-analysis". <i>Inf. Softw. Technol.</i> , vol. 51-7, Jul-2009, pp. 1110-1122.
E25	Fronza, I.; Sillitti, A. ; Succi, G.; Vlasenko, J. "Analysing the usage of tools in pair programming sessions". In: XP, 2011,11p.
E26	Fronza, I.; Sillitti, A.; Succi, G.; Vlasenko, J. "Understanding how novices are integrated in a team analysing their tool usage". In: International Conference on Software and Systems Process, 2011, pp. 204-207
E27	Gallis, H.; Arisholm, E.; Dyba, T. "An initial framework for research on pair programming". In: Proceedings. International Symposium on Empirical Software Engineering, 2003, pp. 132- 142.
E28	Giri, M.; Dewangan, M. "A Study of Pair Programming in the Context of Facilitating the Team Building". In: Second International Conference on Advanced Computing & Communication Technologies (ACCT), 2012, vol., no., pp.20-23, 7-8 Jan. 2012
E29	Han, L; Wenjuan X. "An experimental research of the pair programming in java programming course," In: International Conference on e-Education, Entertainment and e-Management (ICEEEE), 2011, pp.257-260.
E30	Hanks, B. "Distributed pair programming: An empirical study". In:XP, 2004, 12p.
E31	Hanks, B. "Empirical evaluation of distributed pair programming". <i>International Journal of Human Computer Studies</i> , vol. 66-7, Out-2007, pp:530–544.
E32	Hanks, B. "Problems encountered by novice pair programmers". <i>ACM Journal on Educational Resources in Computing</i> , vol. 7-4, Jan-2008.
E33	Hanks, B. "Student attitudes toward pair programming".In: SIGCSE Bull, 2006, pp.113-117.
E34	Hanks, B."Student performance in CS1 with distributed pair programming". In: SIGCSE Bull, 2005, pp. 316-320.
E35	Hanks, B; Matt Brandt."Successful and unsuccessful problem solving approaches of novice programmers". In: SIGCSE Bull., 2009, pp. 24-28.
E36	Hannay, J.; Arisholm, E.; Engvik, H.; Sjoberg, D." Effects of personality on pair programming". In:IEEE Transactions on Software Engineering, vol. 36-1, Fev-2010, pp.61–80.

E37	Ho, C.; Raha, C.; Gehringer,E.;Williams, L. "Sangam: a distributed pair programming plug-in for eclipse". In: OOPSLA, 2004, pp. 73–77.
E38	Höfer, A. "Video analysis of pair programming". In: Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral, 2008, pp. 37-41.
E39	Höfer, A. "Exploratory comparison of expert and novice pair programmers". In: <i>Computing and Informatics</i> , vol. 29-1, Out-2009, pp. 73–91.
E40	Hulkko,H; Abrahamsson , P." A multiple case study on the impact of pair programming on product quality". In: Proceedings of the 27th international conference on Software engineering (ICSE '), 2005, pp. 495-504.
E41	Lui, K.; Chan, K. "Software process fusion by combining pair and solo programming". <i>IET Software</i> , vol. 2-4, Ago-2008, pp. 379–390.
E42	Lui, K.; Chan, K. "Pair programming productivity: Novice-novice vs. expert-expert". <i>International Journal of Human Computer Studies</i> , vol. 64-9, Set-2006, pp.915–925.
E43	Lui, K.; Chan, K." Software process fusion: Exploring process relationships". In: IEEE International Conference on Cognitive Informatics, 2008, pp. 181–184. IEEE.
E44	Lui, K.; Chan, K. "A Cognitive Model for Solo Programming and Pair Programming". In: Proceedings of the Third IEEE International Conference on Cognitive Informatics, 2004, pp.94-102.
E45	Mcdowell, C.; Werner, L. ;Bulock,H. ; Fernald, J. "The effects of pair-programming on performance in an introductory programming course". In: SIGCSE technical symposium on Computer science education, 2002, pp. 38–42.
E46	Mendes, E.; Al-Fakhri, L.; Luxton-Reilly, A. "A replicated experiment of pair-programming in a 2nd-year software development and design computer science course". In: SIGCSE Bull, 2006, pp. 108-112
E47	Mendes,E.; Basil, L.; Fakhri,A.; Reilly, A." Investigating pair-programming in a 2-year software development and design computer science course".In: SIGCSE Bull., 2005, pp. 296–300.
E48	Muller, M.M.; Padberg, F. "An empirical study about the feelgood factor in pair programming". In: Proceedings. of 10th International Symposium on Software Metrics, 2004,pp. 151- 158..
E49	Müller,M. "Do programmer pairs make different mistakes than solo programmers?". <i>J. Syst. Softw.</i> , vol.80-9, Set-2007, pp.1460–1471.
E50	Mustafa, A.; Darroch, F.; M, Toleman. "A framework for understanding the factors influencing pair programming success". In: Proceedings of the 6th international conference on Extreme Programming and Agile Processes in Software Engineering (XP'05), 2005, 10p.
E51	Nagappan, N.; Williams, L., Ferzli, M.; Wiebe, E.; Miller, K.; Balik, S. "Improving the CS1 experience with pair programming". In: SIGCSE Bull, 2003, pp. 359-362.
E52	Navaphan, K.; Gehringer, E.; Culp, J.; Gyllstrom, K., Stotts, D. "Next-generation DPP with Sangam and Facetop". In: Proceedings of the 2006 OOPSLA workshop on eclipse technology eXchange, 2006, pp. 6-10.
E53	Natsu, H.; Favela, J.; Moran, A.L.; Decouchant, D.; Martinez-Enriquez, A.M. "Distributed pair programming on the Web," .In: Proceedings of the Fourth Mexicam International Conference on Computer Science, 2003, pp. 81- 88.
E54	Padberg, F.; Mulle, M. M. "Analyzing the cost and benefit of pair programming". In: Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717), 2003, pp. 166–177.

E55	Padberg, F.; Muller, M.M. "Modeling the impact of a learning phase on the business value of a pair programming project," In: Software Engineering Conference, 2004, pp. 142- 149.
E56	Plonka, L.; Segal, J.; Sharp, H.; Van Der Linden, J. In:"Collaboration in pair programming: Driving and switching". In:XP, 2011, pp. 43-59.
E57	Radermacher, A.; Walia, G. "Investigating student-instructor interactions when using pair programming: An empirical study ".In: 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T), 2011 , pp.41-50.
E58	Radermacher, A.;Walia, G.; Rummelt, R. "Assigning student programming pairs based on their mental model consistency: an initial investigation". In: Proceedings of the 43rd ACM technical symposium on Computer Science Education,2012, pp. 325-330.
E59	Radermacher,A;Walia, G. "Investigating the effective implementation of pair programming: an empirical investigation". In: SIGCS, 2011, pp. 655–660.
E60	Ramli, N.; Fauzi S. "The effects of pair programming in programming language subject". In: International Symposium on Information Technology, 2008, pp.1-4.
E61	Rosen, E.; Salinger, S.; Oezbek, C. "Project Kick-off with Distributed Pair Programming". In: Workshop of Psychology of Programming Interest Group, Madrid, 2010, 15p.
E62	Salleh, N.; Mendes, E.; Grundy, J. "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review". <i>IEEE Transactions on Software Engineering</i> , vol.37-4, Jul-Ago. 2011, pp. 509–525.
E63	Salleh, N.; Mendes, E.; Grundy, J. "The effects of openness to experience on pair programming in a higher education context," In: Software Engineering Education and Training (CSEE&T), 2011, pp.149-158.
E64	Salleh, N.; Mendes, E.; Grundy, J.; Burch, G.S.J. "An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model," In: ACM/IEEE 32nd International Conference on Software Engineering, 2010, pp.577-586.
E65	Salleh, N.; Mendes, E.; Grundy, J.; Burch, G. 2009. "An empirical study of the effects of personality in pair programming using the five-factor model". In: International Symposium on Empirical Software Engineering and Measurement, 2009, 12p.
E66	Salleh, N.; Mendes, E.; Grundy, J.; Burch, G. "The effects of neuroticism on pair programming: an empirical study in the higher education context". In: International Symposium on Empirical Software Engineering and Measurement, 2010,10 p.
E67	Schindler, C. "Agile Software Development Methods and Practices in Austrian IT-Industry: Results of an Empirical Study". In: International Conference on Computational Intelligence for Modelling Control & Automation, 2008, pp.321-326.
E68	Schümmel, T; Lukosch, S." Understanding tools and practices for distributed pair programming".In: Journal of Universal Computer Science, 15(16):3101–3125, 2010.
E69	Sfetsos, P.; Stamelos,I.; Deligiannis,I. "An experimental investigation of personality types impact on pair effectiveness in pair programming". In: Empirical Software Engineering, 2009, pp. 187 - 226.
E70	Shaw, A. "Extending the Pair Programming Pedagogy to Support Remote Collaborations in CS Education". In: Information Technology: New Generations, 2009, pp.1095-1099.
E71	Simon, B;Hanks, B. "First-year students' impressions of pair programming in CS1".J. Educ. Resour. Comput., vol. 7-4, January 2008.

E72	Sison, R. "Investigating the Effect of Pair Programming and Software Size on Software Quality and Programmer Productivity". In: Software Engineering Conference, 2009, pp.187-193.
E73	Slaten, K.M.; Droujkova, M.; Berenson, S.B.; Williams, L.; Layman, L. "Undergraduate student perceptions of pair programming and agile software methodologies: verifying a model of social interaction," In: Proceedings of Agile Conference, 2005, pp. 323- 330.
E74	Stapel, K; Knauss, E.; Schneider,K.; Becker,M." Towards understanding communication structure in pair programming". In: XP, 2010, 15p.
E75	Stotts, D.; Smith, J.; Gyllstrom, K. "Support for distributed pair programming in the transparent video facetop",In: XP, 2004, 15p.
E76	Stotts, D.; Williams, L.; Nagappan, N.; Baheti, P. Jackson. "Virtual teaming: Experiments and experiences with distributed pair programming", Technical report, 2003.
E77	VanDeGrift, T. "Coupling pair programming and writing: learning about students' perceptions and processes".In: SIGCSE Bull., 2004, pp.2-6.
E78	Vanharen, J.; Lassenius, C. "Perceived Effects of Pair Programming in an Industrial Context". In: EUROMICRO Conference on Software Engineering and Advanced Applications, 2007, pp.211-218.
E79	Vanharen, J.; Lassenius, C. "Effects of pair programming at the development team level: an experiment". In: Empirical Software Engineering, 2005, pp. 17-18.
E80	Vanharen, J.; Lassenius, C.; Mantyla, M.V. "Issues and Tactics when Adopting Pair Programming: A Longitudinal Case Study". In: Software Engineering Advances, 2007, pp. 25-31.
E81	Vanharen, J; Korpi, H. "Experiences of Using Pair Programming in an Agile Project" . In: Annual Hawaii International Conference, 2007, pp.274b.
E82	Walle, T.; Hannay, J.E. "Personality and the nature of collaboration in pair programming". In: Empirical Software Engineering and Measurement, 2009, pp.203-213.
E83	Werner, L. "Pair programming strategies for middle school girls". In: Proceedings of the Seventh IASTED International Conference Computers and Advanced Technology in Education, 2004, pp 16-18.
E84	Werner, L ;Hanks, B; McDowell, C." Pair-programming helps female computer science students". <i>J. Educ. Resour. Comput.</i> , vol. 4-1, Mar-2004.
E85	Wernick, P.; Hall, T. "The impact of using pair programming on system evolution a simulation-based study". In: Proceedings of IEEE International Conference on Software Maintenance, 2004, pp. 422- 426.
E86	Wiebe, E; Williams, L; Petlick, J; Balik, S; Miller, C; Ferzli, M. "Pair programming in introductory programming labs" In: American Society for Engineering Education Annual Conference & Exposition, 2003, 12p.
E87	Williams, L; Shukla, A; Anton, A. "An Initial Exploration of the Relationship Between Pair Programming and Brooks' Law". In: Proceedings of the Agile Development Conference (ADC '04), 2004,pp. 11-20.
E88	Williams, L. "Lessons learned from seven years of pair programming at north carolina state university". In:SIGCSE Bull., 2007, pp. 79-83.
E89	Williams, L."Integrating pair programming into a software development process". In: Proceedings of 14th Conference on Software Engineering Education and Training, 2001, pp.27-36.
E90	Williams, L.; Layman, L.; Osborne, J.; Katira, N. "Examining the compatibility of student pair programmers". In: Agile Conference, 2006, pp. 23-28.

E91	Williams, L.; McCrickard, D.S.; Layman, L.; Hussein, K."Eleven Guidelines for Implementing Pair Programming in the Classroom". In: AGILE '08. Conference 2008, pp.445-452.
E92	Williams, L.; McDowell, C.; Nagappan, N.; Fernald, J.; Werner, L.; , "Building pair programming knowledge through a family of experiments,". In: Proceedings of International Symposium on Empirical Software Engineering, 2003,pp. 143- 152.
E93	Xu, S.; Rajlich, V. "Pair Programming in Graduate Software Engineering Course Projects," In: Proceedings 35th Annual Conference Frontiers in Education, 2005, pp. 19-22.
E94	Xu, S; Chen, X. "Pair programming in software evolution". In: Canadian Conference on Electrical and Computer Engineering, 2005, pp.1846-1849.
E95	Zacharis,N. "Measuring the effects of virtual pair programming in an introductory programming java course".IEEE Transactions on Education, vol. 54-, Fev-2011, pp. 168–170.

APÊNDICE C

PROTOCOLO PARA ESTUDO DE CAMPO

Protocolo de Estudo de Caso: Programação em Par Distribuída no Desenvolvimento de Software

Objetivo

Identificar como a Programação em Par Distribuída (PPD) está sendo utilizada no desenvolvimento de software e em que situações poderiam ser adequadas para os profissionais e organizações onde será desenvolvido o estudo de campo.

Característica-chave do método de pesquisa

Este é um roteiro para uma entrevista semi-estruturada com questões abertas. O objetivo é identificar aspectos, benefícios, dificuldades e boas práticas para Programação em Par Distribuída.

Unidades de análise

Projetos de organizações de desenvolvimento distribuído de software (DDS), que utilizam a Programação em Par Distribuída.

Organização desse Protocolo

O protocolo será organizado conforme segue:

1. Procedimentos

a. Levantamento das questões e estruturação do guia para a entrevista

Participantes	Bernardo Estácio
Data	Novembro de 2012
Local	FACIN PUCRS – Faculdade de Informática da PUCRS

b. Revisão do guia para a entrevista

Participantes	Prof. Dr. Rafael Prikladnicki
Data	Novembro de 2012
Local	AGT PUCRS - Agência de Gestão Tecnológica da PUCRS

c. Autorização das empresas participantes

Data	Dezembro de 2012
Local	Porto Alegre

d. Validação de face e conteúdo

Participantes	Taisa Novello
Data	Dezembro de 2012
Local	HP– PUCRS

e. Pré-teste

Participantes	Roni Orsoletta – Mestrando
Data	Dezembro de 2012
Local	FACIN PUCRS – Faculdade de Informática da PUCRS

f. Aplicação das entrevistas

Data	Dezembro de 2012
Local	Escritórios das empresas, Skype

2. Escolha das pessoas entrevistadas

Respondentes:

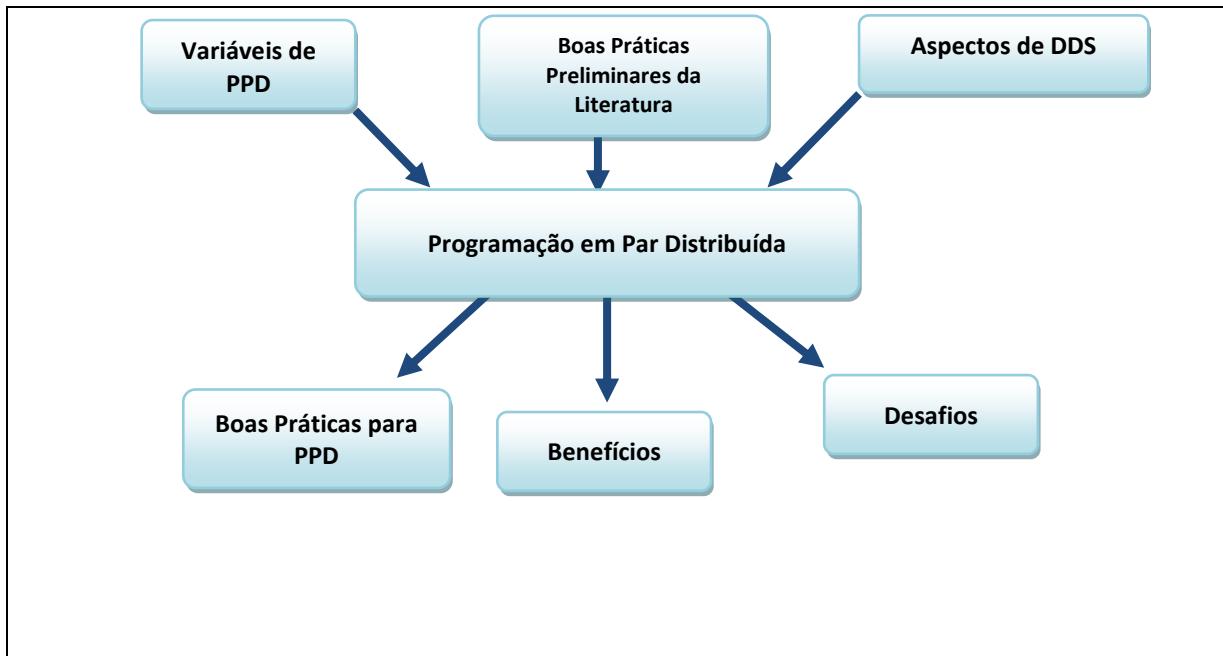
- a. Desenvolvedores que utilizam programação em par distribuída

3. Outros recursos utilizados

- a. Recursos tecnológicos
 - i. Microsoft Excel e Word
- b. Recursos materiais
 - i. Uma sala de reuniões reservada
 - ii. Um gravador para gravar as entrevistas
 - iii. Papel e caneta

4. Modelo de estudo e dimensões da pesquisa

O esquema a seguir representa graficamente os principais aspectos focados no desenvolvimento deste trabalho. O esquema foi baseado nos resultados de uma revisão sistemática realizada pelos autores.



- **Variáveis de PPD:** representam as medidas utilizadas nos estudos analisados da revisão sistemática para avaliar a prática de PPD.
- **Boas Práticas Preliminares da Literatura:** um conjunto de estratégias para PPD, identificadas em uma revisão sistemática.

5. **Aspectos de DDS:** indicam os benefícios, desafios e configurações de equipes no contexto do DDS que podem influenciar na prática de PPD.

6. Coleta de dados

Entrevistas semiestruturadas com questões abertas.

7. Análise de dados

Após a transcrição das entrevistas será realizada uma análise dos dados coletados.

Questões		
Dados Demográficos	Nome:	Idade: ____ anos.
	Curso (nível mais alto):	
	Instituição:	Concluído em:
	Tempo de experiência profissional com desenvolvimento de software:	____ anos.
	Tempo de experiência profissional com DDS:	____ anos
	Tempo de experiência profissional com Mét. Ágeis:	____ anos
	Tempo de experiência profissional com PP	____ anos
	Tempo de experiência profissional com PPD	____ anos
	Departamento/área:	
	Vínculo empregatício:	Tempo de empresa: ____ anos.
Função atual:		

Questões	
Aspectos de DDS	1. Como é utilizado o DDS na sua organização? Em qual nível de distribuição (local, nacional, continental ou global)?
	2. Quais os benefícios verificados a partir da utilização do DDS?
	3. Quais as dificuldades encontradas que estão relacionadas ao DDS? Como essas foram contornadas?

Questões	
Características do projeto	4. Quais as características do projeto em que foi utilizada a PPD ? <ul style="list-style-type: none"> • Tamanho de projeto (componentes, total de horas, etc.); • Tamanho da equipe (gerentes, analistas, testadores, etc.); • Conhecimento técnico e experiência dos envolvidos com PP e PPD; • Fuso-horário; • Idioma;
	5. Em que momento e quais foram os motivos que levaram a adoção pela empresa da PPD no desenvolvimento deste projeto?

Questões	
Variáveis	6. Que tipo de efeito a PPD gerou na qualidade do código? Por quê? Quais foram as estratégias usadas para alcançar este efeito?
	7. Que tipo de efeito a PPD gerou na produtividade da equipe? Por quê? Quais foram estratégias usadas para alcançar este efeito?
	8. Que tipo de efeito a PPD gerou na comunicação da equipe? Por quê? Quais foram estratégias usadas para alcançar este efeito?
	9. Que tipo de efeito a PPD gerou na diferença de conhecimento entre os pares? Por quê? Quais foram estratégias usadas para alcançar este efeito?
	10. Além da qualidade do código, da produtividade, comunicação e diferença de conhecimento existe alguma outra variável que gera um efeito ao utilizar PPD? Qual a variável e que tipo de efeito? Quais foram estratégias usadas para alcançar este efeito?

Questões	
Característica de PPD	11. Existe alguma diretriz organizacional para a utilização da prática?
	12. Qual a infraestrutura, métodos que foram utilizados com a PPD?
	13. Quais as ferramentas (software) que são usadas para PPD? Existe alguma ferramenta de desenvolvimento que é específica para PPD?
	14. Existe algum líder (coach) que facilita a adoção da prática na organização?
	15. Existe algum critério estabelecido para a formação dos pares que adotam PPD?
	16. Quem é o responsável pela escolha dos pares?
	17. A diferença de conhecimento e experiência pares é um empecilho no contexto distribuído? Por quê?
Questões	
Benefícios	18. Em sua opinião, com relação à transferência de conhecimento entre os envolvidos foram verificados benefícios a partir da utilização da PPD? Quais?
	19. Em sua opinião, com relação ao tempo de execução da tarefa entre os envolvidos foram verificados benefícios a partir da utilização da PPD? Quais?
	20. Em sua opinião, com relação ao esforço entre os envolvidos foram verificados benefícios a partir da utilização da PPD? Quais?
	21. Em sua opinião, com relação à motivação entre os envolvidos foram verificados benefícios a partir da utilização da PPD? Quais?
	22. Quais outros benefícios foram obtidos a partir da utilização da PPD no desenvolvimento de software?
Questões	
Desafios	23. Em sua opinião, quanto à comunicação quais foram os desafios encontrados na PPD e como esses foram solucionados?
	24. Em sua opinião, quanto à colaboração quais foram os desafios encontrados na PPD e como esses foram solucionados?
	25. Quais outras dificuldades foram enfrentadas para realizar a PPD? E como essas foram contornadas?
Questões	
Opinião	26. Tendo por base a sua experiência profissional, como você compara o desenvolvimento realizado com e sem o uso de PPD?
	27. A partir da experiência vivenciada em projetos que utilizam a PPD, quais seriam suas sugestões visando complementar ou melhorar o ambiente já existente?

English Version

Question		
Demographic data	Name:	Age:
	Course:	
	Institution:	Class:
	Professional experience with software development: _____	
	Professional experience with DSD : _____	
	Professional experience with Agile Methods: _____	
	Professional experience with PP _____	
	Professional experience with DPP _____	
	Department:	
	Current Function:	Company time: _____

Questions	
DDS Aspects	1. How does your organization use DSD ? In which distribution level (local, national, continental, global)?
	2. What are the benefits obtained by using DSD?
	3. Which from the difficulties found are related to DSD? How were they handled?

Questions	
Project Characteristics	4. What are the characteristics of the project in which DPP was used? project size (people, total hours, etc.); team size (managers, analysts, testers, etc.); technical knowledge and expertise of those involved with PP and DPP; time zone; language;
	5. In which moment and what were the reasons that lead the company to adopt DPP on this project development?

Questions	
Variables	6. What kind of effect did DPP bring to the quality of code? Why? What were the strategies used to reach this effect?
	7. What kind of effect did DPP bring to the team productivity? Why? What were the strategies used to reach this effect
	8. What kind of effect did DPP bring to the team communication? Why? What were the strategies used to reach this effect?
	9. What kind of effect did DPP bring to the knowledge disparity between pairs? Why? What were the strategies used to reach this effect?
	10. Beyond the quality of code, productivity, communication and knowledge disparity is there any other variable affected by the use of DPP? Which variables and what are the effects observed? What were the strategies used to reach these effects

Questions	
PPD Characteristic	11. Is there any company guideline for using DPP?
	12. What infrastructure and methods have been used with DPP?
	13. What tools are used for DDP? Is there any development tool specific to DPP?
	14. Is there a facilitator or leader (coach) to support this practice in the company?
	15. Is there any criterion established for arranging the pairs in DPP?
	16. Who is responsible for choosing the pairs?
	17. The difference of knowledge and experience is a problem? Why?

Questions	
Benefits	18. In your opinion, regarding the knowledge transfer between the pairs, were there benefits from the use of DPP? Which?
	19. In your opinion, regarding the task execution time, were there benefits from the use of DPP? Which?
	20. In your opinion, regarding the effort between the pairs, were there benefits from the use of DPP? Which?
	21. In your opinion, regarding the motivation, were there benefits from the use of DPP? Which?
	22. Have you seen other benefits from the use of DPP? Which?

Questions	
Challenges	23. In your opinion, what were the communication challenges found in DPP and how were they solved??
	24. In your opinion, what were the collaboration challenges found in DPP and how were they solved?
	25. What other challenges were identified in DPP? And how were they solved?

Questions	
Opinion	26. Based on your experience, how do you compare the development performed with and without the use of DPP?
	27. From your experience with projects using DPP, which would be your suggestions to complement the existing environment