



Pós-Graduação em Ciência da Computação

Karla Michele Barbosa da Silva

## MÉTODO PARA CRIAÇÃO DE REQUISITOS COMPLETOS NO PROCESSO DE DESENVOLVIMENTO DE *SOFTWARE*



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
<http://cin.ufpe.br/~posgraduacao>

Recife  
2019

Karla Michele Barbosa da Silva

**MÉTODO PARA CRIAÇÃO DE REQUISITOS COMPLETOS NO PROCESSO DE  
DESENVOLVIMENTO DE *SOFTWARE***

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

**Área de Concentração:** Engenharia de *software*.

**Orientador:** Fábio Queda Bueno da Silva

Recife  
2019

Catálogo na fonte  
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S586m Silva, Karla Michele Barbosa da  
Método para criação de requisitos completos no processo de desenvolvimento de software / Karla Michele Barbosa da Silva. – 2019.  
133 f.: il., fig., tab.

Orientador: Fábio Queda Bueno da Silva.  
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2019.  
Inclui referências e apêndices.

1. Engenharia de software. 2. Requisitos de software. I. Silva, Fábio Queda Bueno da (orientador). II. Título.

005.1

CDD (23. ed.)

UFPE- MEI 2019-100

**Karla Michele Barbosa da Silva**

**“Método para Criação de Requisitos Completos no Processo de Desenvolvimento de Software”**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 14/03/2019.

**BANCA EXAMINADORA**

---

Prof. Dr. André Luís de Medeiros Santos  
Centro de Informática/UFPE

---

Prof. Dr. Cleviton Vinicius Fonsêca Monteiro  
Departamento de Estatística e Informática/ UFRPE

---

Prof. Dr. Fabio Queda Bueno da Silva  
Centro de Informática/UFPE  
**(Orientador)**

*Dedico este trabalho a Deus porque dEle e por Ele, e para Ele, são todas as coisas.*

## AGRADECIMENTOS

A Deus, acima de tudo, pelo dom da vida, pelo privilégio de poder ser considerada filha do Altíssimo e pela oportunidade de avançar mais um degrau na minha trajetória acadêmica. De tal modo que me faltam palavras para expressar toda a minha gratidão pela sabedoria, força e saúde concedida para enfrentar cada desafio dessa jornada.

Ao meu pai, Carlos Roberto (*in memoriam*), que infelizmente não teve a oportunidade de participar da conclusão desse sonho, mas que sempre foi um grande apoiador. Permanentemente será meu exemplo de esforço e caráter. A minha mãe, Neide Cordeiro, por sempre ter uma palavra de apoio que muitas vezes me motivou a continuar nessa jornada, mesmo diante das dificuldades. Talvez sem perceber me ensinava sobre humildade e bondade. Da mesma forma, gostaria de agradecer a minha irmã, Thays Silva, pelo companheirismo e amizade de sempre.

Aos meus familiares, em especial aos meus avós, que sempre torceram por mim e apoiaram as minhas escolhas. Da mesma forma, agradeço aos meus amigos e colegas de trabalho que me apoiaram mesmo durante os momentos de ausência durante o desenvolvimento dessa pesquisa.

Ao meu orientador Fábio Queda, por acreditar no meu trabalho, mas também sou grata por seus ensinamentos e críticas que levarei durante toda a minha trajetória acadêmica. Agradeço também aos membros do grupo de pesquisa *Human Aspects in Software Engineering* (HASE) pelos *feedbacks* recebidos durante os encontros. Do mesmo modo, não poderia deixar de expressar minha gratidão ao meu ex-orientador, Alexandre Mota, por me guiar nos primeiros momentos da vida de mestrandia. Além deles, reconheço que tenho uma dívida incalculável com todos os meus ex-professores, os quais foram peças fundamentais na construção de uma base sólida para que eu estivesse aqui hoje.

Ao Programa de Pós-Graduação em Ciência da Computação do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE) pela oportunidade recebida. Agradeço também a empresa parceira da universidade pela confiança e disponibilidade em apresentar os problemas e colaborar com a solução que foi planejada durante o período da pesquisa. Sem vocês, essa pesquisa não teria acontecido.

Em resumo, agradeço a todos que participaram dessa etapa da minha vida. Muitas vezes, com simples palavras de motivação ou mesmo oferecendo ajuda no que fosse possível. Talvez vocês não saibam, mas vocês me deram forças para não desistir da caminhada. Sou muito grata por tudo!

## RESUMO

Os requisitos de um projeto de *software* são itens significativos durante todo o ciclo de desenvolvimento do projeto. Eles, geralmente, são apresentados na literatura relacionados as principais causas de sucesso, falha ou atraso de projetos. Por isso, ter requisitos com qualidade é o desejo de muitas equipes e gerentes, mas é comum encontrar na literatura estudos empíricos que relatam problemas e desafios relacionados a ausência de características que deveriam estar presentes em um bom requisito. Em especial, problemas relacionados com a falta de completude dos mesmos. O objetivo principal dessa dissertação foi construir, verificar e validar um método para que equipes de desenvolvimento, ao aplicá-lo ao longo do ciclo de vida de um projeto, possam produzir requisitos de *software* que possuam todas as informações relevantes incluídas, ou seja, sejam completos. Entrevistas semi-estruturadas foram conduzidas com colaboradores de uma empresa parceira da universidade com o intuito de auxiliar o diagnóstico do problema. Também foi realizada uma revisão sistemática da literatura para que ações mencionadas em estudos primários pudessem ser aplicadas na pesquisa. E por último, foi desenvolvida uma pesquisa-ação, em conjunto com os membros de uma equipe de desenvolvimento de *software*, para a construção, verificação e validação de um método para produzir requisitos completos. Esta pesquisa gerou como resultado o método CReq-SW, que poderá ser utilizado para a produção de requisitos de *software* completos. O mesmo foi construído, verificado e validado durante a pesquisa por uma equipe de desenvolvimento de sistemas. Além disso, pode-se citar como resultado secundário a importância das cerimônias de refinamento de requisitos, validação do cliente nos testes de aceitação e os benefícios da escrita de testes para capturar requisitos. Adicionalmente, através da revisão sistemática da literatura, foi possível identificar uma lista de ações que podem ser colocadas em prática em qualquer projeto de desenvolvimento para tornar os requisitos mais explícitos. Essa pesquisa contribuiu com o esclarecimento sobre a possibilidade de um time pode tornar os requisitos mais completos sem a intervenção do cliente, e esses achados podem fundamentar pesquisas e aplicações futuras do método. Além disto, o processo descrito para a aplicação do método CReq-SW poderá ser utilizado para guiar outros trabalhos que optarem por utilizar a pesquisa-ação como método de pesquisa.

**Palavras-chaves:** Desenvolvimento Ágil de Software. Requisitos de *Software*. Pesquisa-ação.

## ABSTRACT

Requirements of a software project are significant items throughout the entire project development life cycle. Usually, they are presented in the literature related to the main causes of success, failure, or delay of projects. This might be the reason why good requirements are the desire of many teams and managers; however, it is still common to find empirical studies in the literature that report problems and challenges related to the absence of certain characteristics that should be present in a good requirement. In particular, problems related to their lack of completeness. The main goal of this dissertation was to build, verify and validate a method that development teams can apply it through the entire project life cycle to produce requirements with all relevant data included. In other words, complete requirements. A semi-structured interview was conducted with the employees of a company that has a partnership with the university. The goal of those interviews was to help diagnose the problem. Also, a systematic literature review was conducted, so that actions cited in the primary studies could be applied in the research. Coupled with, an action-research was carried out with the members of the development team to build, verify and validate the method to produce complete requirements. This research produce as a result the method CReq-SW which can be used to create complete software requirements. The method was built, verified and validated by a software development team during this research. In addition, the importance of product backlog refinement ceremony, customer validation in acceptance tests, and the benefits of writing tests to gather requirements can be cited as a secondary result. In the same way, through the systematic literature review, it was possible to identify a list of actions that can be adopted by any development project to make the requirements more explicit. This research contributed to the academy and industry providing shreds of evidence regarding the possibility to have complete requirements without the customer intervention. Besides that, the process described for the application of the method can be used as a guideline to others who choose the action-research as a research method.

**Keywords:** Agile Software Development. Software Requirements. Action Research.

## LISTA DE FIGURAS

Figura 1 – Modelo Cascata e Ágil Apresentado por Kukhnavets (2016) . . . . .	26
Figura 2 – Ciclo da Pesquisa-ação . . . . .	31
Figura 3 – Relação entre a Rec Tecnologia, UFPE, <i>GlobeTech Institute - Brazil</i> (GTI-Br) e <i>GlobeTech</i> . . . . .	40
Figura 4 – Organograma geral da empresa . . . . .	41
Figura 5 – Estrutura do Projeto B12 . . . . .	42
Figura 6 – Linha do Tempo com as Etapas e Principais Atividades da Pesquisa- ação - Parte 1 . . . . .	44
Figura 7 – Linha do Tempo com as Etapas e Principais Atividades da Pesquisa- ação - Parte 2 . . . . .	45
Figura 8 – Atividades da Etapa de Instalação da Pesquisa . . . . .	48
Figura 9 – Organograma Parcial com Destaque para os Participantes Entrevistados	50
Figura 10 – Atividades da Etapa de Diagnóstico . . . . .	51
Figura 11 – Exemplo de Transcrição do Áudio das Entrevistas . . . . .	54
Figura 12 – Exemplo de Codificação . . . . .	55
Figura 13 – Atividades da Etapa de Planejamento . . . . .	56
Figura 14 – Possíveis Quadrantes para a Pesquisa-ação . . . . .	59
Figura 15 – Atividades da Etapa de Implementação . . . . .	63
Figura 16 – Relacionamento entre os Componentes do Método CReq-SW . . . . .	72
Figura 17 – Linha do Tempo para a Ação de Refinamento de Requisitos . . . . .	78
Figura 18 – Exemplo da <i>Matrix</i> de Rastreabilidade do Projeto B12 . . . . .	82
Figura 19 – <i>String</i> de busca . . . . .	101
Figura 20 – Exemplo da Extração . . . . .	103
Figura 21 – Fluxograma das Etapas de Seleção das Empresas e Instalação da Pes- quisa . . . . .	107
Figura 22 – Fluxograma das Atividades 1 e 2 (parcial) da Etapa de Diagnóstico . .	108
Figura 23 – Fluxograma das Atividades 2, 3, 4, 5 e 6 da Etapa de Diagnóstico . . .	109
Figura 24 – Fluxograma da Etapa de Planejamento . . . . .	110
Figura 25 – Fluxograma das Etapas de Implementação e Avaliação . . . . .	111

## LISTA DE TABELAS

Tabela 1 – Cronograma das Etapas da Pesquisa-ação . . . . .	45
Tabela 2 – Números Gerais sobre os Artefatos Gerados Durante a Pesquisa . . . .	47
Tabela 3 – Práticas Encontradas na Literatura para Tornar os Requisitos Mais Completos . . . . .	57
Tabela 4 – Motivação para a não seleção das ações . . . . .	61
Tabela 5 – Descrição do Processo do Método CReq-SW . . . . .	74
Tabela 6 – Instrumentos Utilizados Durante a Aplicação do Método CReq-SW . .	75
Tabela 6 – Instrumentos Utilizados Durante a Aplicação do Método CReq-SW . .	76
Tabela 7 – Participantes e suas Responsabilidades na Aplicação do Método CReq- SW . . . . .	76
Tabela 8 – Ações Escolhidas Relacionadas com a Participação do Cliente . . . . .	84
Tabela 9 – Problemas Comuns na Engenharia de Requisitos em Contextos Ágeis .	96
Tabela 10 – Palavras-chaves e Sinônimos . . . . .	102

## LISTA DE QUADROS

Quadro 1 – Divisão das Atividades por Iteração . . . . .	63
--	----

## LISTA DE ABREVIATURAS E SIGLAS

<b>ACs</b>	<i>Acceptance criteria</i>
<b>CIn</b>	Centro de Informática
<b>CReq-SW</b>	<i>Complete Requirements for Software Projects</i>
<b>GPS</b>	<i>Global Positioning System</i>
<b>GTI-Br</b>	<i>GlobeTech Institute - Brazil</i>
<b>HASE</b>	<i>Human Aspects in Software Engineering</i>
<b>IPI</b>	Imposto sobre Produtos Industrializados
<b>NaPiRE</b>	<i>Naming the Pain in Requirements Engineering</i>
<b>P&amp;D</b>	Pesquisa e Desenvolvimento
<b>PO</b>	<i>Product Owner</i>
<b>QAW</b>	<i>Quality Attribute Workshop</i>
<b>RH</b>	Recursos Humanos
<b>RSL</b>	Revisão Sistemática da Literatura
<b>TDD</b>	<i>Test-Driven Development</i>
<b>UFPE</b>	Universidade Federal de Pernambuco
<b>UML</b>	<i>Unified Modeling Language</i>
<b>XP</b>	<i>eXtreme Programming</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	MOTIVAÇÃO E JUSTIFICATIVA	15
1.2	CONTEXTO DO DESENVOLVIMENTO DA PESQUISA	16
1.3	OBJETIVOS DA PESQUISA	16
1.4	ESTRUTURA DO TRABALHO	17
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>18</b>
2.1	ENGENHARIA DE REQUISITOS	18
2.1.1	<b>Características de Bons Requisitos</b>	<b>19</b>
2.1.2	<b>Tipos de Requisitos</b>	<b>20</b>
2.1.3	<b>Principais Atividades</b>	<b>22</b>
2.1.4	<b>Cascata <i>versus</i> Ágil</b>	<b>26</b>
2.1.5	<b>Problemas Comuns</b>	<b>26</b>
2.2	PESQUISA-AÇÃO	28
2.2.1	<b>Origem da Pesquisa-ação</b>	<b>28</b>
2.2.2	<b>Características da Pesquisa-ação</b>	<b>29</b>
2.2.3	<b>Fases da Pesquisa-ação</b>	<b>30</b>
2.2.4	<b>Pesquisa-ação <i>versus</i> Consultoria</b>	<b>32</b>
2.2.5	<b>Rigor Científico</b>	<b>32</b>
2.3	TRABALHOS RELACIONADOS	33
<b>3</b>	<b>MÉTODO DE DESENVOLVIMENTO DA PESQUISA</b>	<b>36</b>
3.1	QUESTÕES DA PESQUISA	36
3.2	PREMISSAS DA PESQUISA	36
3.3	ESTRATÉGIAS DA PESQUISA	37
3.3.1	<b>Unidade de Análise</b>	<b>37</b>
3.3.2	<b>Metodologia</b>	<b>37</b>
3.3.3	<b>Método de Abordagem</b>	<b>37</b>
3.3.4	<b>Propósito</b>	<b>38</b>
3.3.5	<b>Método de Pesquisa</b>	<b>39</b>
3.3.6	<b>Técnicas de Coleta e Análise de Dados</b>	<b>39</b>
<b>4</b>	<b>AMBIENTE DE DESENVOLVIMENTO DA PESQUISA</b>	<b>40</b>
4.1	VISÃO GERAL DOS AMBIENTES DA PESQUISA	40
4.1.1	<b>Dados demográficos da empresa</b>	<b>41</b>
4.2	CARACTERIZAÇÃO DO PROJETO	42

4.2.1	<b>Processo de Desenvolvimento</b>	42
<b>5</b>	<b>DESENVOLVIMENTO DA PESQUISA-AÇÃO</b>	<b>44</b>
5.1	SELEÇÃO DA EMPRESA	47
5.2	INSTALAÇÃO DA PESQUISA	47
5.3	DIAGNÓSTICO	50
5.4	PLANEJAMENTO	56
5.5	IMPLEMENTAÇÃO	62
5.6	AVALIAÇÃO	65
<b>6</b>	<b>RESULTADOS</b>	<b>67</b>
6.1	RESULTADOS TEÓRICOS DA PESQUISA-AÇÃO	67
<b>6.1.1</b>	<b>Revisão Sistemática da Literatura</b>	<b>67</b>
<b>6.1.2</b>	<b>O Método</b>	<b>71</b>
6.1.2.1	Componentes do Método	71
<b>6.1.2.1.1</b>	<b>Objetivo</b>	<b>72</b>
<b>6.1.2.1.2</b>	<b>Premissas e Limitações</b>	<b>73</b>
<b>6.1.2.1.3</b>	<b>Processo</b>	<b>73</b>
<b>6.1.2.1.4</b>	<b>Instrumentos</b>	<b>75</b>
<b>6.1.2.1.5</b>	<b>Participantes</b>	<b>76</b>
6.2	RESULTADOS EMPÍRICOS DA PESQUISA-AÇÃO	77
<b>6.2.1</b>	<b>Ações Implementadas</b>	<b>77</b>
6.2.1.1	Reunião de Refinamento de Requisitos ( <i>Backlog Refinement</i> )	78
6.2.1.2	<i>Sprint Review</i>	80
6.2.1.3	Participação do Cliente na Escrita ou Revisão dos Testes de Aceitação	80
6.2.1.4	Rastreabilidade entre Requisitos e Testes	81
6.2.1.5	Escrita de testes para capturar requisitos	82
<b>6.2.2</b>	<b>O Poder da Intervenção do Cliente</b>	<b>83</b>
<b>7</b>	<b>CONCLUSÃO</b>	<b>86</b>
7.1	LIMITAÇÕES E AMEAÇAS A VALIDADE DA PESQUISA	87
7.2	TRABALHOS FUTUROS	88
	<b>REFERÊNCIAS</b>	<b>90</b>
	<b>APÊNDICE A – PROBLEMAS COMUNS NA ENGENHARIA DE REQUISITOS</b>	<b>96</b>
	<b>APÊNDICE B – METODOLOGIA USADA NA REVISÃO SISTEMÁTICA DA LITERATURA</b>	<b>101</b>

APÊNDICE C – <i>FRAMEWORK</i> PARA PESQUISA-AÇÃO NA INDÚSTRIA . . . . .	107
APÊNDICE D – ROTEIRO DAS ENTREVISTAS . . . . .	112
APÊNDICE E – SLIDES DA APRESENTAÇÃO 1 - PROBLEMAS COMUNS . . . . .	115
APÊNDICE F – SLIDES DA APRESENTAÇÃO 2 - DIAGNÓSTICO	117
APÊNDICE G – SLIDES DA APRESENTAÇÃO 3 - APRESENTAÇÃO DAS POSSÍVEIS AÇÕES . . . . .	125
APÊNDICE H – SLIDES DA APRESENTAÇÃO 4 - AÇÕES ESCOLHIDAS . . . . .	128
APÊNDICE I – SLIDES DA APRESENTAÇÃO 5 - <i>FEEDBACK</i> .	133

# 1 INTRODUÇÃO

Este capítulo descreve, na seção 1.1, a motivação e justificativa para a condução deste trabalho. Na seção seguinte, é apresentado de forma resumida, algumas características do contexto do desenvolvimento dessa pesquisa. Já na seção 1.3, são expostos os objetivos geral e específicos do trabalho. E por fim, a seção 1.4 descreve como esta dissertação está estruturada.

## 1.1 MOTIVAÇÃO E JUSTIFICATIVA

A publicação e adoção do Manifesto Ágil (BECK et al., 2001) no início do século XXI trouxe diversas implicações sobre como as equipes de desenvolvimento de *software* lidavam com requisitos. Anteriormente, era necessário um esforço considerável na fase de planejamento para que todos os requisitos do projeto fossem definidos antes das outras atividades começarem. Por isso, a publicação do Manifesto Ágil é considerada uma quebra de paradigmas nesse quesito. Os autores do Manifesto argumentavam que esse planejamento não deveria ser feito todo no início do projeto, mas sim diluído ao longo do ciclo de vida do projeto (BECK et al., 2001).

Ao todo, o Manifesto Ágil (BECK et al., 2001) apresenta quatro valores e doze princípios. No entanto, os itens abaixo, um valor e quatro princípios, respectivamente, merecem atenção especial devido ao foco dessa pesquisa.

- *Software* em funcionamento mais que documentação abrangente;
- Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de *software* com valor agregado;
- Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento;
- Entregar frequentemente *software* funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
- *Software* funcionando é a medida primária de progresso;

Contudo, durante essa corrida para entregar soluções funcionando, muitos times acabam enfrentando alguns desafios geralmente relacionados a documentação, verificação ou até mesmo ao processo de engenharia de requisitos.

Por isso, muitas pesquisas vêm sendo desenvolvidas com o intuito de endereçar os desafios relacionados a engenharia de requisitos comumente enfrentados pelas equipes de desenvolvimento de *software*. Por exemplo, pesquisas com engenheiros de *software* e de testes demonstram que muitas vezes os requisitos repassados para a equipe estão incompletos ou possuem requisitos “escondidos” (FERNÁNDEZ et al., 2016), e esse é apenas um dos inúmeros desafios citados na literatura. Por outro lado, declarações claras de requisitos são apontadas como o

terceiro fator para o sucesso de um projeto (The Standish Group, 1994). Mas, como tornar os requisitos de um projeto de *software* mais completos para um time? Ou seja, como tornar todas as informações relevantes mais explícitas? Esse é o problema selecionado como objeto de estudo dessa pesquisa.

O intuito de abordar essa questão gerando condições para que ações possam ser planejadas, implementadas e avaliadas na indústria focando na mudança do ambiente estudado, é a principal motivação para o desenvolvimento dessa pesquisa.

## 1.2 CONTEXTO DO DESENVOLVIMENTO DA PESQUISA

Esta pesquisa foi realizada em uma empresa de desenvolvimento de *software* através de uma cooperação entre a indústria e a universidade. Essa parceria existe desde 2015 e por questões de confidencialidade, não será usado o nome real da empresa. Daqui em diante, a empresa onde a pesquisa foi desenvolvida será chamada por Rec Tecnologia.

A Rec Tecnologia trabalha com desenvolvimento de projetos de *software*, recebe as demandas dos seus projetos de um instituto do cliente localizado em outro estado brasileiro, e é adepta de práticas ágeis (mais detalhes sobre o ambiente de pesquisa serão apresentados no capítulo 4). A empresa em questão é a “dona do problema” anteriormente apresentado e irá trabalhar colaborativamente para que todo o ciclo da pesquisa-ação ocorra no contexto onde a pesquisa foi aplicada.

## 1.3 OBJETIVOS DA PESQUISA

Esta pesquisa tem como objetivo principal desenvolver um método para propor, analisar e validar ações que podem ser implementadas em projetos de desenvolvimento de *software* com o intuito de tornar os requisitos completos para todos os envolvidos no processo. O termo “requisitos completos” refere-se a uma das qualidades necessária para um bom requisito, e está relacionada a especificação de todas as condições que podem ser aplicáveis a um requisito. Ou seja, não devem existir requisitos definidos nas “entrelinhas”. Mais detalhes sobre essa característica serão apresentados na subseção 2.1.1.

Além disso, para que o objetivo principal dessa pesquisa fosse alcançado, os seguintes objetivos específicos foram elencados:

- Construir, aplicar e avaliar ações direcionadas a melhorar o entendimento dos requisitos de *software* pelo time de uma empresa parceira da indústria;
- Contribuir com a comunidade científica com informações empíricas sobre a adoção de práticas e ferramentas utilizadas para o esclarecimento de requisitos de *software*;
- Contribuir com a comunidade científica apresentando um *framework* para condução de pesquisa-ação na indústria.

## 1.4 ESTRUTURA DO TRABALHO

Essa dissertação está estruturada em sete capítulos. Os quais, com exceção do primeiro capítulo que é o capítulo atual, estão detalhados logo em seguida.

No capítulo 2 é apresentada uma revisão da literatura explorando os conceitos de engenharia de requisitos e pesquisa-ação, que é o método de pesquisa no qual esse trabalho foi desenvolvido. Além disso, a seção 2.3 apresenta alguns trabalhos relacionados que foram usados como referência durante o desenvolvimento dessa pesquisa.

No capítulo 3 é descrito o método de desenvolvimento da pesquisa. Em especial, esse capítulo apresenta as questões, as premissas e a estratégia da pesquisa (seção 3.3). Por sua vez, a seção 3.3 divide-se em seis subseções que apresentam detalhes sobre a metodologia, o método de abordagem, as técnicas de coleta de dados e outras informações.

Já no capítulo 4 é retratado o ambiente de desenvolvimento da pesquisa, explicitando as características da empresa e dos projetos selecionados. Além disso, a subseção 4.2.1 fornece uma visão geral sobre o processo de desenvolvimento do projeto estudado na presente pesquisa.

No capítulo 5, o processo de desenvolvimento da pesquisa-ação desenvolvida nesse trabalho é explicado etapa por etapa. Desde a seleção da empresa “dona do problema” até a fase de avaliação dos efeitos da pesquisa.

No capítulo 6 estão descritos os resultados teóricos e empíricos obtidos com a realização da presente pesquisa, incluindo a apresentação e descrição completa do método CReq-SW.

Por fim, o capítulo 7 descreve as considerações finais sobre o trabalho, as limitações, as ameaças a validade e quais as recomendações para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta dois conceitos importantes para um melhor entendimento do trabalho desenvolvido nessa pesquisa.

Na seção 2.1 é apresentado o conceito de engenharia de requisitos, as características essenciais de bons requisitos, como são classificados na literatura, quais as principais atividades, as diferenças entre os modelos cascata e ágil (no contexto de requisitos) e quais são os problemas comumente citados na literatura. Em seguida, a seção 2.2 é dedicada ao método de pesquisa que será aplicado nesse trabalho enfatizando suas origens, características e etapas. E por último, foram listados os principais trabalhos associados a essa pesquisa.

### 2.1 ENGENHARIA DE REQUISITOS

Sem dúvidas, entregar um projeto de *software* que atenda às necessidades dos usuários é um tema recorrente em pesquisas na área de engenharia de *software* e em palestras na indústria (KOTONYA; SOMMERVILLE, 1998). Por exemplo, para alguns engenheiros de *software*, entender os requisitos de um projeto pode ser considerado uma das tarefas mais difíceis entre tantas outras no processo de desenvolvimento do produto (PRESSMAN, 2010).

No entanto, em primeiro lugar, faz-se necessário entender como o termo requisito é definido na literatura, e uma definição comumente usada é a apresentada no glossário da IEEE (1990) que descreve requisitos como:

1. uma condição ou capacidade necessária por um usuário para resolver um problema ou alcançar um objetivo;
2. uma condição ou capacidade que deve ser atingida ou pertencer a um sistema, componente desse sistema, produto ou serviço para satisfazer um acordo, padrão, especificação ou qualquer outro documento formalmente imposto;
3. uma representação documentada de uma condição ou capacidade como citados nos itens (1) ou (2);

Ou seja, os requisitos definem as necessidades e os objetivos de um usuário, as condições e as propriedades do sistema que será desenvolvido e a documentação dos mesmos (POHL, 2010). O autor também usa o termo “artefato dos requisitos” como correspondente ao item 3 da definição apresentada pela IEEE (1990).

Contudo, propor e desenvolver um sistema é uma tarefa bastante complexa, visto que, as atividades relacionadas aos requisitos permeiam por todo o processo de desenvolvimento de um produto. Segundo Kotonya e Sommerville (1998), o termo “engenharia de requisitos” foi inventado justamente para cobrir todas essas atividades relacionadas aos requisitos, e para

o autor, o uso do termo engenharia implica no uso sistemático e repetitivo de técnicas para garantir que os requisitos do sistema estão completos, consistentes e relevantes.

Nuseibeh e Easterbrook (2000) vão além e afirmam que o objetivo da engenharia de requisitos consiste em um conjunto de atividades direcionadas a elicitaco, anlise, e especificaco de requisitos que de forma no ambgua devem refletir o propsito do *software* conectando e alinhando as necessidades do usurio, do cliente, e de outras partes afetadas pelo sistema.

Outra definio de engenharia de requisitos comumente apresentada na literatura  o contedo apresentado por Zave (1997). Em seu artigo, a autora enfatiza que a engenharia de requisitos  o ramo da engenharia de software que se preocupa basicamente com trs coisas para atingir os objetivos apresentados pelo mundo real. So elas:

- Investigao de objetivos, funoes e restrioes de uma aplicao de software;
- Especificao precisa de software;
- Gerenciamento da evoluo e das famlias do software;

Embora possua mais de duas dcadas, essa citao ainda  muito usada. Visto que existe um destaque para os objetivos do mundo real, especificao precisa e evoluo.

### 2.1.1 Caractersticas de Bons Requisitos

Como j mencionado anteriormente, os requisitos de uma aplicao esto diretamente relacionados ao propsito da mesma. Por isso, o desenvolvimento de bons requisitos  essencial para que o produto possa atingir os objetivos para o qual foi planejado. Por isso, IEEE (1998), Hull, Jackson e Dick (2004), Robertson e Robertson (2006) e Pohl (2010) listam alguns critrios que um requisito deve possuir para ser considerado um bom requisito. De forma resumida, algumas das qualidades de um requisito so apresentadas abaixo:

- **Completo:** A declarao do requisito precisa incluir todas as informaoes que podem ser relevantes para algum *stakeholder*. Sejam elas relacionadas a funcionalidades, performance, ou qualquer outra restrio. A ttulo de exemplo, considere o seguinte requisito:

Req. 1: O sistema deve cobrar uma multa caso o aluno entregue o livro em atraso.

O requisito acima no indica o valor da multa, nem se ela  proporcional aos dias atrasados ou se  apenas uma taxa fixa de atraso, por exemplo.

- **Rastrevel:** Deve ser possvel seguir o ciclo de vida de um requisito desde a sua origem at o seu uso. A rastreabilidade pode ser til para entender decisoes tomadas previamente, por exemplo.

- **Não ambíguo:** Deve haver apenas uma interpretação do requisito. É preciso ser cuidadoso ao optar pelo uso da linguagem natural para documentação de requisitos, já que ela é passível de ambiguidade.
- **Compreensível:** Cada declaração de requisito deve ser entendida facilmente por todos os *stakeholders*.
- **Consistente:** Requisitos não podem entrar em contradição uns com os outros. Como exemplo, considere os dois requisitos abaixo:

Req. 1: O sistema deve aceitar pagamentos através do Paypal.

Req. 2: Apenas pagamentos com cartão de crédito devem ser aceitos.

Nesse caso, provavelmente um dos requisitos acima será alterado ou excluído para que eles não apresentem informações conflitantes.

- **Verificável:** Os *stakeholders* do projeto devem ser capazes de checar se a implementação do sistema atende ao que foi documentado. Ou seja, os critérios de aceitação precisam ser claros e objetivos.
- **Atômico:** Um requisito deve descrever um simples e coerente fato. Múltiplos elementos isolados devem ser quebrados em outros requisitos.

É importante ressaltar que alguns desses critérios podem ser usados tanto para requisitos individuais como para um grupo de requisitos.

### 2.1.2 Tipos de Requisitos

Levemente diferente de outros materiais (KOTONYA; SOMMERVILLE, 1998; SOMMERVILLE, 2011) que basicamente classificam os requisitos em funcionais e não funcionais, Robertson e Robertson (2006) diferenciam os requisitos em três principais tipos: requisitos funcionais, requisitos não funcionais e restrições.

#### Requisitos Funcionais

Como o próprio nome já diz, requisitos funcionais especificam as funcionalidades que o sistema deve prover para seus usuários, pessoas ou outros sistemas. Sommerville (2011) vai além e afirma que requisitos funcionais também descrevem como o sistema deve reagir ao receber determinadas entradas de dados e como o mesmo deve se comportar em situações particulares. Ou até mesmo, o que o sistema não deve fazer. Para ilustrar, o seguinte requisito funcional é apresentado pelo autor:

Req. 1: Um usuário de um *smartphone* com o *Global Positioning System* (GPS) habilitado deve ser capaz de pesquisar por locais próximos a ele.

### Requisitos não funcionais

Requisitos não funcionais definem qualidades que um produto deve ter ou o quão bem feito o produto faz as coisas que deveria fazer. Geralmente, eles são responsáveis por fazerem o produto ser atrativo, usável, rápido, seguro, entre outros (ROBERTSON; ROBERTSON, 2006). Sem dúvida, eles exercem um papel fundamental no sucesso de um projeto.

Para exemplificar, o autor apresenta o seguinte exemplo de requisito “não funcional” relacionado a usabilidade:

Req. 1: O produto deve ser fácil de usar.

No entanto, esse requisito pode ser muito vago e idealístico a princípio, e poderia ser refinado para algo menos ambíguo como o exemplo apresentado abaixo:

Req. 1: O produto deve ser fácil de usar por membros do público ainda que os mesmos não leiam Inglês.

Com isso em mente, Pohl (2010) afirma que é preciso ser cauteloso por que a maioria dos requisitos que são classificados como não funcionais são na verdade requisitos funcionais mal especificados, e alguns poucos são requisitos de “qualidade” de verdade. Por isso, que o autor optou por chamá-los de requisitos de “qualidade”, ao invés de requisitos não funcionais. No entanto, para essa pesquisa, foi usada a classificação proposta por Robertson e Robertson (2006) juntamente com as características definidas anteriormente por Zave (1997).

### Restrições

Baseado em Robertson e Robertson (2006), Pohl (2010) define restrições como um requisito organizacional ou tecnológico que restringe a forma que o sistema deve ser desenvolvido. No entanto, essas restrições podem ser de diferentes origens e impactar diferentes sujeitos. Por exemplo, quanto a origem, as restrições podem ser de fatores culturais, organizacionais, físicos e outras. Sobre as partes impactadas, o próprio sistema, ou o seu processo de desenvolvimento podem ser alvos dessas limitações. O exemplo abaixo é um caso de restrição que afeta o processo de desenvolvimento (POHL, 2010).

Req. 1: O sistema deve ser desenvolvido usando a metodologia Extreme Programming (XP).

Para Gause (2005), cada restrição tem um impacto sobre as alternativas para a implementação dos requisitos funcionais e de qualidade, visto que alternativas vão sendo limitadas conforme as restrições.

### 2.1.3 Principais Atividades

Como já mencionado anteriormente, as atividades relacionadas a engenharia de requisitos estão presentes em todo o processo de desenvolvimento. Segundo Pohl (2010), essas atividades podem ser classificadas de duas formas: atividades *core* ou transversais (termo original: *cross-sectional*).

As atividades do *core* contribuem significativamente para que os objetivos da engenharia de requisitos sejam atingidos. São elas: documentação, elicitacão e negociação. Já as atividades transversais, a saber, validação e gerenciamento, influenciam o processo de engenharia de requisitos, servem como suporte para as atividades do *core*, e protegem os resultados da engenharia de requisitos.

É importante lembrar que essas atividades não necessariamente vão ser executadas uma após a outra. Geralmente, elas vão se sobrepor dependendo da organização ou das necessidades do projeto (KOTONYA; SOMMERVILLE, 1998; PRESSMAN, 2010; POHL, 2010).

#### Documentação

Para Pohl (2010), a atividade de documentação tem como objetivo documentar informações importantes elicitadas ou desenvolvidas enquanto uma outra atividade *core* ou transversal estava sendo executada. De acordo com Sommerville e Sawyer (1997) e Kovitz (1998), é indispensável a documentação dos requisitos por vários motivos:

- É uma forma de persistir informação;
- É uma referência comum para todos os *stakeholders*;
- Pode ser considerado como um estímulo a comunicação e discussões sobre o conteúdo do documento;
- É mais objetiva do que a comunicação verbal e menos propenso a alterações durante o diálogo;
- Pode ser usado como suporte para treinar novos colaboradores;
- Ajuda na disseminação do conhecimento evitando que o mesmo fique concentrado em indivíduos específicos;
- Ajuda a refletir sobre o problema. Quando alguém está documentando os requisitos, essa pessoa está refletindo sobre o problema.

Os requisitos podem ser documentados usando uma linguagem natural ou qualquer outra forma que seja entendível por todos os outros *stakeholders*. Contudo, é importante que eles possuam as características de um bom requisito que já foram apresentadas na subseção 2.1.1.

### Elicitação

O termo elicitación de requisitos sugere que o processo consiste simplesmente em uma transferência entre o que foi elicitado pelo engenheiro de requisitos e o conhecimento do cliente. No entanto, essa não é a realidade na prática. Geralmente, os clientes não têm uma ideia clara dos requisitos ou talvez diferentes pessoas na organização tenham ideias diferentes dos requisitos (KOTONYA; SOMMERVILLE, 1998; PRESSMAN, 2010). De acordo com Pohl (2010), os objetivos dessa atividade são:

- Identificar fontes relevantes de requisitos;
- Elicitar requisitos existentes das fontes já identificadas;
- Desenvolver novos e inovadores requisitos;

É importante ressaltar que os requisitos podem surgir das mais diversas fontes. Por exemplo, ideias, modelos de requisitos, sistemas já existentes, entre outros. E por isso, as pessoas responsáveis por elicitación dos requisitos precisam usar diferentes técnicas para descobrir toda essa informação. Algumas técnicas comumente citadas na literatura são: entrevistas, *workshops*, observações e outras (KOTONYA; SOMMERVILLE, 1998; POHL, 2010).

### Negociação

Segundo Pohl (2010), a fase de negociação tem como objetivo identificar conflitos, analisar suas causas, resolve-los usando estratégias apropriadas e documentar a resolução dos mesmos. No entanto, Pressman (2010) esclarece que essa não é uma atividade que se deve buscar ganhadores ou perdedores. Para o autor, o acordo realizado precisa ser útil para ambas as partes. Por isso, essa técnica é comumente conhecida como *win-win*. Além dessa, outra prática citada por Kotonya e Sommerville (1998) e Pohl (2010) é a matriz de interações, cujo objetivo é visualizar e documentar sobreposição de requisitos e seus conflitos.

Outra coisa que deve ser lembrada é que conflitos sobre os requisitos podem surgir durante qualquer atividade do processo de engenharia de requisitos, e os mesmos podem ser de diferentes tipos. Por exemplo, em um determinado projeto pode haver conflito de dados, valores ou interesse (EASTERBROOK, 1994; POHL, 2010). O quadro abaixo apresenta um exemplo de conflito de valores.

Um *stakeholder* solicitou que o cadastro do usuário deve possuir integrações com o Facebook<sup>a</sup>, Twitter<sup>b</sup> e Github<sup>c</sup>.

Um outro *stakeholder* é contra as integrações com redes sociais e acredita que isso não é importante.

<sup>a</sup><facebook.com>

<sup>b</sup><twitter.com>

<sup>c</sup><github.com>

Para Kotonya e Sommerville (1998), reuniões são a forma mais efetiva de negociar requisitos e resolver conflitos. Segundo o autor, cada conflito deve ser discutido individualmente e a decisão tomada para um não deve ser assumida como verdade para os demais.

### Validação

Sommerville (2011) define que a atividade de validação de requisitos é o processo no qual é feita a verificação se os requisitos propostos atendem as demandas do cliente. Antes de mais nada, é interessante ressaltar que os termos validação e verificação são comumente confundidos na literatura e indústria. Por isso, Robertson e Robertson (2006), Dzida e Freitag (1998) e Pohl (2010) argumentam que a verificação está relacionada a checar o sistema em busca de erros. Geralmente, a verificação está associada com a seguinte pergunta: “nós estamos construindo o sistema da forma correta?”. Já a validação, por está relacionada as necessidades do cliente, se preocupa em responder “nós estamos construindo o sistema certo?”. Sendo assim, para a atividade em questão, o foco será em validação.

Por ser uma atividade *cross-sectional*, a validação começa na checagem de artefatos, passa pela validação da execução das atividades e se estende até checar se as considerações do contexto estão sendo observadas. Por isso, Pohl (2010) sugere alguns princípios:

- **Envolver os *stakeholders* corretos:** os *stakeholders* responsáveis por validar os requisitos precisam estar envolvidos no processo de desenvolvimento. No entanto, eles podem ser internos ou externos a organização;
- **Separar a detecção de defeitos da correção dos mesmos:** inicialmente, a detecção de defeitos pode ser priorizada sobre a correção visando uma maior concentração de esforços no entendimento do defeito. Acima de tudo, é importante avaliar cada defeito com relação a sua criticidade e esforço necessário para ser corrigido.
- **Ter diferentes visões independentes:** visões diferentes são úteis na detecção de diferenças no entendimento. Contudo, é primordial que essas diferenças sejam detectadas e consolidadas.
- **Usar formatos de documentação apropriados:** geralmente, nem todos os formatos de documentação são apropriados para validar todos os tipos de requisitos. Por exemplo,

casos de usuário escritos em linguagem natural podem ser úteis para validar funcionalidades relacionadas aos usuários do sistema, mas podem não ser interessantes para validar requisitos de arquitetura.

- **Criação de artefatos de desenvolvimento durante a validação:** durante a validação dos requisitos, os mesmos são intensivamente explorados. As informações obtidas durante essa atividade podem dar origem a artefatos de teste ou até mesmo a um manual de usuário.
- **Validação repetitiva:** conhecimento sobre os requisitos vão sendo adquiridos ao longo do projeto. Por este motivo, é importante sempre validar novamente os requisitos para garantir que os mesmos continuam válidos.

Mesmo assim, não é difícil encontrar erros de requisitos em sistemas já entregues, e a correção dos mesmos é muito custosa. A saber, a correção para alguns podem custar até 100 vezes mais quando comparados com um erro de programação (KOTONYA; SOMMERVILLE, 1998). Por isso, Kotonya e Sommerville (1998) sugerem que, se possível, esses erros devem ser descobertos durante a fase de validação de requisitos. Com isto em mente, Kotonya e Sommerville (1998), Sommerville (2011) e Pohl (2010) sugerem algumas técnicas que podem ser usadas na validação. Por exemplo, inspeção, prototipação, revisão de requisitos, geração de casos de teste e outras. Mais detalhes de algumas dessas técnicas serão apresentados na seção 5.4.

## Gerenciamento

A atividade *cross-sectional* de gerenciamento de requisitos consiste em um processo para gerenciar as mudanças de requisitos de um sistema. Isso é necessário por que no início, provavelmente, os *stakeholders* não possuem um entendimento aprofundado sobre as suas necessidades. A medida que o sistema vai sendo construído, esses requisitos vão sendo refinados (KOTONYA; SOMMERVILLE, 1998; SOMMERVILLE, 2011). Essas requisições de mudanças também podem ocorrer devido a alterações em leis, políticas organizacionais, envolvimento de outros *stakeholders*, e etc.

Visando facilitar esse processo de mudanças, alguns autores (KOTONYA; SOMMERVILLE, 1998; PRESSMAN, 2010; POHL, 2010) propõem algumas atividades e práticas que podem ajudar o time a identificar, controlar e rastrear os requisitos e suas mudanças durante o processo de desenvolvimento. Algumas das técnicas ou práticas sugeridas são: matriz de rastreabilidade entre os requisitos, estabelecimento de um processo de requisição de mudanças, uso de ferramentas que apoiam a requisição de mudanças, técnicas para priorização de requisitos e outras. No entanto, Kotonya e Sommerville (1998) esclarecem que talvez essa atividade em projetos de pequeno porte tenha características menos formais do que quando executada em um grande projeto.

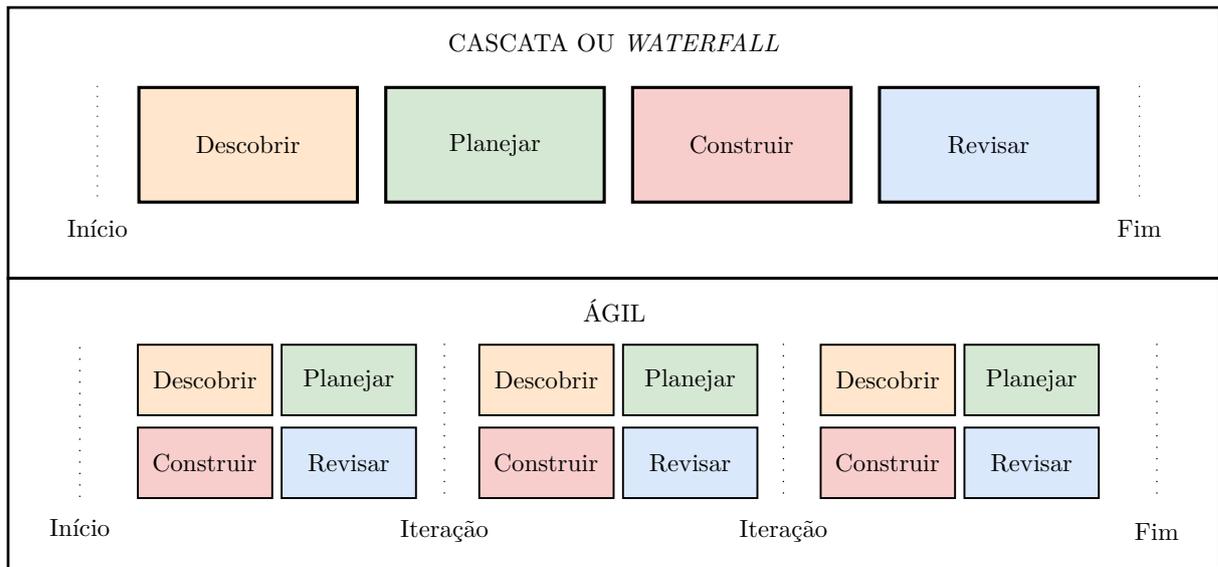


Figura 1 – Modelo Cascata e Ágil Apresentado por Kukhnavets (2016)

#### 2.1.4 Cascata versus Ágil

As atividades descritas na subseção 2.1.3 estão presentes nos modelos de desenvolvimento ágil e cascata (termo original, *Waterfall*). O modelo cascata também é conhecido como o modelo do ciclo de vida clássico. No entanto, para Paetsch, Eberlein e Maurer (2003), no âmbito dos requisitos, a diferença consiste em como as atividades são abordadas em cada um dos modelos.

No modelo *Waterfall*, as atividades são executadas em sequência e os requisitos precisam estar claros antes que a fase de *design* seja iniciada. Além disso, as fases são bem definidas e não existe sobreposição das mesmas. É importante notar que nesse modelo os testes são executados apenas no final do ciclo, e muitas vezes, devido a atrasos no cronograma, essa atividade tem seu período encurtado. (PRESSMAN, 2010; BALAJI; MURUGAIYAN, 2012).

Já no modelo ágil, um dos princípios que o norteia é o “Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo” (BECK et al., 2001). Por isso, todo o ciclo de desenvolvimento precisa ser executado em repetitivas iterações menores. Além disso, Paetsch, Eberlein e Maurer (2003) afirmam que no processo ágil é um pouco mais difícil distinguir onde começa cada atividade, e que algumas vezes elas podem ter sido agrupadas.

A Figura 1 apresenta os dois modelos de desenvolvimento cascata e ágil baseados na figura exposta por Kukhnavets (2016).

#### 2.1.5 Problemas Comuns

Kotonya e Sommerville (1998) acreditam que quando os requisitos de um sistema não estão corretos, é certo que um grande número de consequências irá surgir. Sob o mesmo ponto de vista, o relatório do *The Standish Group* (1994) aponta requisitos incompletos como uma das

principais causas para a falha de projetos. Para Pohl (2010), geralmente, a importância da engenharia de requisitos não é levada a sério como deveria.

Por isso, de forma resumida, os itens abaixo apresentam os principais problemas e desafios encontrados na literatura. Os mesmos são oriundos de 2 revisões sistemáticas (INAYAT et al., 2015; ELGHARIANI; KAMA, 2016), 3 mapeamentos sistemáticos (MEDEIROS et al., 2015; HEIKKILA et al., 2015; CURCIO et al., 2018) e do estudo de Fernández et al. (2016) que não foi incluído em nenhuma das revisões ou mapeamentos, mas apresenta achados relevantes relacionados ao tema. Sendo todos relacionados a engenharia de requisitos. É importante ressaltar que a lista completa de desafios apresentados por cada estudo está disponível no Apêndice A.

- **Documentação insuficiente:** em seus estudos focados em contextos ágeis, Inayat et al. (2015), Medeiros et al. (2015), Elghariani e Kama (2016) e Curcio et al. (2018) citam a falta ou carência de documentação como um problema resultante de um processo de engenharia de requisitos deficiente.
- **Falta de suporte do cliente:** essa queixa foi citada em todos os trabalhos avaliados (INAYAT et al., 2015; MEDEIROS et al., 2015; ELGHARIANI; KAMA, 2016; HEIKKILA et al., 2015; FERNÁNDEZ et al., 2016; CURCIO et al., 2018). Para Inayat et al. (2015), a realidade desse item vai de encontro ao que é defendido pelos métodos ágeis, visto que os mesmos encorajam a cooperação com o cliente (BECK et al., 2001). De acordo com a literatura, também é necessário que o cliente tenha competência para tomar decisões e que exista consenso quando mais de um cliente está envolvido no processo (CURCIO et al., 2018).
- **Problemas com estimativas:** Todos os trabalhos analisados mencionam algum tipo de problema com estimativas. Sejam elas relacionadas a esforço, orçamento ou cronograma (INAYAT et al., 2015; MEDEIROS et al., 2015; ELGHARIANI; KAMA, 2016; HEIKKILA et al., 2015; FERNÁNDEZ et al., 2016; CURCIO et al., 2018).
- **Arquitetura inadequada:** Para Inayat et al. (2015), Elghariani e Kama (2016) e Curcio et al. (2018), a arquitetura escolhida durante o início dos ciclos de desenvolvimento talvez seja considerada inadequada quando outros requisitos aparecerem. Por isso, eles enfatizam o custo do *refactoring*. Medeiros et al. (2015) também comenta que as definições de arquitetura não são escaláveis por causa das constantes mudanças inerentes do ambiente ágil.
- **Falta de preocupação com requisitos não funcionais:** Alguns autores (INAYAT et al., 2015; MEDEIROS et al., 2015; ELGHARIANI; KAMA, 2016) alegam que a falta de preocupação com os requisitos que representam as qualidades do produto é outro item que também merece atenção. Muitas vezes, esses requisitos só aparecem tarde demais.
- **Requisitos incompletos ou escondidos:** Fernández et al. (2016), em seu estudo, lista requisitos incompletos ou escondidos como um dos problemas mais citados em um *survey*

realizado com 228 organizações. De forma semelhante, Medeiros et al. (2015) apontam requisitos não estáveis, superficiais e ambíguos como um desafio para os projetos no contexto ágil.

- **Falhas de comunicação:** Medeiros et al. (2015), Elghariani e Kama (2016) e Fernández et al. (2016) citam falhas de comunicação como um problema relevante de engenharia de requisitos. É importante ressaltar que nesse item são considerados todos os tipos de falha de comunicação. Por exemplo, entre o time e o cliente, ou até mesmo problemas de comunicação entre os membros do time de desenvolvimento do projeto.
- **Mudanças:** Mudanças nos objetivos, processos de negócio e/ou requisitos também são problemas populares na literatura (INAYAT et al., 2015; MEDEIROS et al., 2015; ELGHARIANI; KAMA, 2016; FERNÁNDEZ et al., 2016).

É interessante mencionar que existem outros artigos que tratam de problemas em engenharia de requisitos, mas os mesmos não foram incluídos por que já foram citados em uma das revisões ou mapeamentos.

## 2.2 PESQUISA-AÇÃO

A pesquisa-ação foca especialmente na combinação entre a teoria e a prática. Por isso, antes de qualquer passo, faz-se necessário atingir a pré-condição básica para esse tipo de pesquisa: a necessidade de ter um “dono do problema”, que esteja disposto a participar colaborativamente de todo o processo. Em alguns casos, o “dono do problema” e o pesquisador são a mesma pessoa. Só é importante lembrar que o proprietário do problema passa a ser um colaborador da pesquisa (BASKERVILLE; WOOD-HARPER, 1996; DAVISON; MARTINSONS; KOCK, 2004; EASTERBROOK et al., 2008).

Outra característica importante da pesquisa-ação é a de propor e implantar mudanças em situações previamente estudadas do mundo real com o objetivo de melhorá-las, e não apenas observá-las como outros métodos de pesquisa geralmente o fazem. Por isso, a pesquisa-ação é vista como um processo ativo, onde as convicções são redefinidas a medida que os resultados vão sendo coletados.

### 2.2.1 Origem da Pesquisa-ação

Na literatura, o termo “pesquisa-ação” (em inglês, *Action-Research*) aparece pela primeira vez no artigo do professor Lewin (1947) no contexto da psicologia social associada a grupos minoritários da sociedade. Em seu trabalho, Lewin discute sobre mudanças de hábitos alimentares em um cenário pós segunda guerra mundial. É importante ressaltar que a necessidade de criar uma ponte entre o concreto e o abstrato era um tema recorrente nos artigos de Lewin. Peters e Robinson (1984) também mencionam essa necessidade como uma lacuna que existe entre as ações e as teorias sociais.

Alguns anos depois, na clínica britânica *Tavistock*, Trist (1976, p.43-58) escreve sobre o método de pesquisa-ação desenvolvido durante sua pesquisa, que tem como objetivo estudar transtornos psicológicos e sociais no âmbito militar. Pouco depois, a pesquisa-ação começou a ser conhecida em outros contextos; por exemplo, o trabalho de Checkland (1981) na universidade de *Lancaster* é considerado um marco na literatura de sistema de informação. Além do ramo tecnológico, a pesquisa-ação foi amplamente utilizada em pesquisas das áreas de saúde e administração.

Olhando para o mundo da engenharia de software, na primeira década do século XXI, os trabalhos de Easterbrook et al. (2008) e Sjoberg, Dyba e Jorgensen (2007) afirmam que a pesquisa-ação faz parte dos métodos de pesquisa empírica mais relevantes para a engenharia de software.

### 2.2.2 Características da Pesquisa-ação

Entretanto, não existe um consenso na literatura sobre as características da pesquisa-ação. Provavelmente, devido aos diferentes contextos e domínios nos quais a pesquisa é conduzida. Contudo, Suassuna (2017) em sua tese, sumarizou as principais características da pesquisa-ação. Esses atributos são oriundos de outros trabalhos que usaram essa forma de pesquisa como formação de conhecimento para as organizações. As principais peculiaridades da pesquisa-ação são:

- A pesquisa-ação precisa ser uma experiência que acontece no mundo real, cujas as ações propostas precisam ser percebidas como fatos por todos os envolvidos.
- Esse método de pesquisa atua fortemente na relação entre o pesquisador e as pessoas envolvidas da organização. Juntos eles trabalham na identificação dos problemas e buscam uma provável solução.
- Existe um esforço na fundamentação de um embasamento teórico sólido para que a implementação das ações seja mais confiável e consistente. Por isso, qualquer ferramenta, técnica, modelos ou métodos desenvolvidos podem ser considerados artefatos resultantes da pesquisa.
- Durante a pesquisa-ação, o pesquisador está interessado em vivenciar em um contexto do mundo real, a aplicação de *framework* teórico. Com isso, o mesmo tem como objetivo receber *feedback* dessa vivência, alterar a teoria com os resultados desse parecer e rodar o ciclo novamente. Cada iteração desse ciclo, pode acrescer ricamente a teoria.
- É um procedimento destinado a gerar uma transformação verdadeira nos grupos envolvidos. Fazem parte desse procedimento, os objetivos definidos por qualquer uma das partes, ou por um acordo entre ambos.

- É necessário que existam normas e maneiras de observar, coletar, controlar e avaliar os dados e conclusões oriundos da pesquisa.
- Os dados obtidos através das entrevistas, discussões e observações devem ser considerados como uma fotografia da organização como um todo e não apenas uma amostra.
- Os grupos envolvidos na pesquisa precisam participar ativamente das discussões sobre o problema e as pessoas da organização devem ter um diálogo aberto com os pesquisadores.
- É flexível, visto que a pesquisa-ação vai sendo desenhada ao passo que a mesma evolui. Ou seja, a princípio, o pesquisador não tem conhecimento sobre o modo que a pesquisa deverá ser conduzida para atingir os objetivos previamente definidos.
- É moldável, uma vez que oferece suporte aos envolvidos na pesquisa para tornar mais fácil a incorporação do conhecimento no dia a dia da organização.
- Critérios claros precisam ser estabelecidos antes do início da pesquisa, também devem ser definidas formas de gerenciar a alteração desses critérios durante o ciclo da pesquisa-ação para que a mesma não deixe de ser uma pesquisa-ação.
- A primeira fase da pesquisa-ação corresponde ao diagnóstico que recomenda que os dados sejam levantados para auxiliar a identificação do problema. Futuramente, os mesmos serão usados na construção das ações a serem implementadas na organização.
- É orientada ao futuro, pois lida com as preocupações das pessoas visando a criação de um amanhã mais desejado pela organização.

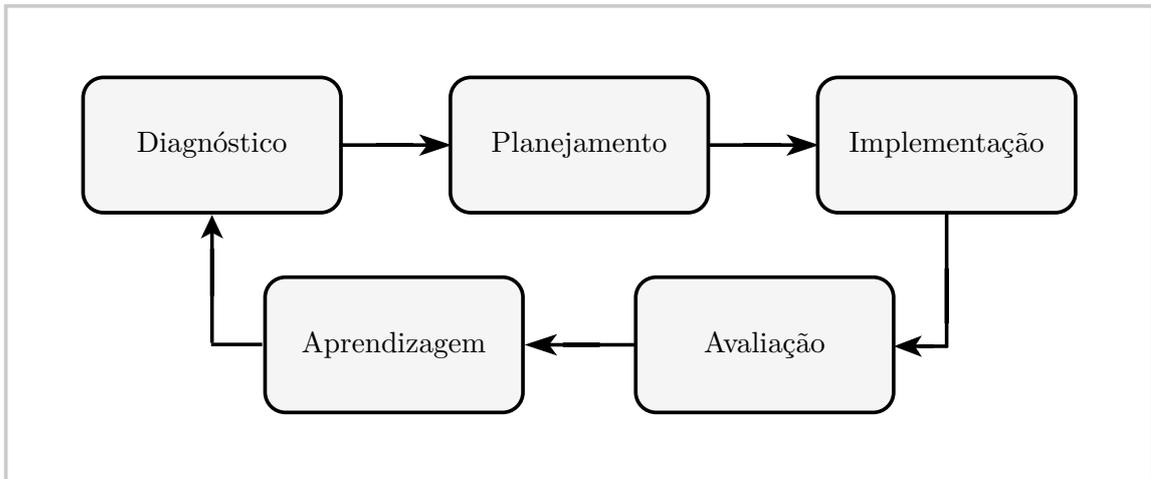
### 2.2.3 Fases da Pesquisa-ação

Para Susman e Evered (1978), a pesquisa-ação é composta de cinco fases que compõe um modelo cíclico. A Figura 2 apresenta um processo aprimorado do modelo proposto inicialmente por ambos autores.

Baskerville e Wood-Harper (1996) caracterizam as cinco fases apresentadas na Figura 2 da seguinte forma:

- **Diagnóstico:** Fase de identificação dos problemas primários que motivaram o desejo da organização por mudanças. Isso inclui uma profunda interpretação dos complexos problemas organizacionais, que serão úteis para o desenvolvimento de hipóteses sobre a organização e o contexto do seu problema.
- **Planejamento:** Etapa onde pesquisadores e colaboradores baseados em um *framework* teórico elaboram um plano com as ações que devem ser aplicadas na empresa proprietária do problema.

Figura 2 – Ciclo da Pesquisa-ação



**Fonte:** Adaptação do Ciclo de Pesquisa-ação Proposto por Susman (1983)

- **Implementação:** De forma colaborativa, pesquisadores e membros da organização trabalham na intervenção e implementação das ações definidas na etapa anterior. A estratégia de intervenção pode ser diretiva, não diretiva ou tática. As duas primeiras são relacionadas a como o pesquisador direciona as mudanças. Já na estratégia tática, por exemplo, pode ser contratada uma terceira pessoa especialista na ação a ser implementada.
- **Avaliação:** Depois de todas as ações terem sido executadas, é necessário avaliar os efeitos teóricos das mesmas. Além disso, pesquisadores e profissionais precisam verificar se a intervenção contribuiu para o alívio ou extinção do problema inicialmente diagnosticado. Se a mudança for positiva, deve-se sempre criticar o contexto e as ações tomadas para garantir que elas foram a razão do sucesso, mas caso os resultados não sejam positivos, o *framework* para um próximo ciclo de pesquisa-ação deverá ser elaborado.
- **Aprendizagem:** Embora seja listado como última fase do ciclo de pesquisa-ação, essa etapa deve permear todas as outras fases apresentadas anteriormente. O conhecimento adquirido durante a pesquisa-ação é muito útil para o pesquisador, mas esse conhecimento pode ser extremamente proveitoso também para a organização detentora do problema e para a comunidade científica.

É importante lembrar que de acordo com Baskerville e Wood-Harper (1996), a pesquisa-ação pode continuar independentemente dos resultados. Seja ela para obter conhecimento sobre a organização ou até mesmo para validar hipóteses.

#### 2.2.4 Pesquisa-ação *versus* Consultoria

Durante o decorrer dessas fases, o pesquisador e a organização estão seriamente imersos no processo de melhoria do problema previamente identificado, e muitas vezes, essa relação acaba sendo confundida com uma consultoria. No entanto, Baskerville (1999) lista 5 características que o autor considera útil na diferenciação dos mesmos.

- **Motivação:** A pesquisa-ação é motivada e embasada na perspectiva científica. Já a consultoria é focada em benefícios comerciais. Provavelmente ligados a lucros ou a obtenção de conhecimento sobre os problemas organizacionais.
- **Comprometimento:** Diferente da consultoria que está comprometida apenas com o cliente, a pesquisa-ação, está comprometida com o cliente e com a comunidade acadêmica para a produção de conhecimento científico.
- **Abordagem:** colaboração é primordial para qualquer pesquisa-ação. Por outro lado, geralmente, a consultoria valoriza muito uma visão externa de um ponto de vista não enviesado para obter uma visão objetiva dos problemas organizacionais.
- **Embasamento para recomendações:** na pesquisa-ação, a fundamentação das ações recomendadas é obtida através de um *framework* teórico. Em contrapartida, os consultores sugerem soluções baseadas em suas próprias experiências anteriores.
- **Essência do entendimento organizacional:** o entendimento organizacional é baseado no modelo iterativo experimental das mudanças que estão acontecendo nas organizações. Em alternativa, os consultores, geralmente, desenvolvem o entendimento deles de forma independente através de análises críticas da situação.

Em resumo, consultores geralmente são contratados de acordo com experiência que eles adquiriram no mercado. Já os pesquisadores da pesquisa-ação buscam usar a teoria de forma experimental para resolver os problemas (BASKERVILLE, 1997).

#### 2.2.5 Rigor Científico

Outra preocupação comumente citada na literatura sobre a pesquisa-ação, diz respeito a rigor. Sem dúvidas, em qualquer trabalho científico é importante entender como garantir rigor e validade, especialmente quando as premissas são extremamente diferentes das comumente usadas pelos positivistas. Por isso, Cunha e Figueiredo (2002) resumem algumas táticas propostas por alguns autores com a finalidade de garantir rigor durante o processo da pesquisa-ação. São elas:

- Um *framework* teórico deve ser definido no início do processo, e novos conhecimentos vão sendo construídos baseados nesse *framework*.

- O uso de ciclos é extremamente recomendado. Durante cada ciclo, o pesquisador deve tentar confrontar os resultados emergentes com o intuito de garantir que eles foram oriundos das ações implementadas e não de outras variáveis do ambiente.
- A metodologia e as perguntas de pesquisa devem ser criteriosamente analisadas e refinadas em cada ciclo.
- A coleta e a interpretação de dados devem fazer parte de todos os ciclos. Isso pode ser útil para comparar ciclos futuros.
- Durante cada ciclo, o pesquisador deve focar apenas nas situações onde as pessoas concordam ou discordam. De forma que todo hábito incomum ou estranho deve ser ignorado.
- Dados divergentes devem ser procurados, pois os mesmos podem ser aproveitáveis para o confronto dos dados. Acrescentando-se que a literatura existente também pode ajudar bastante nesse sentido.
- Diferentes fontes de informação ou diferentes perspectivas sobre determinado assunto precisam ser usadas com o objetivo de descobrir o que é realmente verdade mesmo que isso seja feito através de teorias opostas. Esse objetivo pode ser atingido através de mais de um método de coleta de dados, múltiplos entrevistadores, dados redundantes coletados próximos ao mesmo informante e outros.
- Os resultados das mudanças oriundas da pesquisa podem ser usadas como um recurso adicional de informação para confrontar as teorias emergentes.

Devidos as suas peculiaridades relacionadas a colaboração entre pesquisadores e as organizações, a pesquisa-ação pode ser considerada como uma excelente escolha para pesquisadores que buscam trabalhar nesses dois mundos.

### 2.3 TRABALHOS RELACIONADOS

Nessa seção são apresentados os trabalhos relacionados a essa pesquisa. Basicamente, existem dois estudos (RAMESH; CAO; BASKERVILLE, 2010; FERNÁNDEZ et al., 2016) que foram úteis na caracterização do problema e desenvolvimento da pesquisa.

O trabalho de Ramesh, Cao e Baskerville (2010) apresenta os resultados de uma pesquisa empírica sobre engenharia de requisitos cujos dados são oriundos de 16 empresas de desenvolvimento de software dos Estados Unidos que usam métodos ágeis. Depois de analisar os dados, os autores identificaram as seguintes práticas ágeis:

- Comunicação face a face;
- Priorização extrema;

- Engenharia de requisitos iterativa;
- Protótipos;
- *Review* e testes;
- Planejamento constante.

Por outro lado, Ramesh, Cao e Baskerville (2010) também identificaram 7 desafios que precisam ser endereçados para que as organizações alcancem o sucesso.

- Problemas com estimativas de custo e tempo (principalmente, durante as fases iniciais do projeto);
- Acesso e participação do cliente. O time de desenvolvimento precisa da confiança e disponibilidade do cliente para que a comunicação entre eles seja efetiva;
- Arquitetura inadequada ou imprópria. Escolhas iniciais de arquitetura podem mostrar-se inadequadas quando novos requisitos começam a aparecer;
- Requisitos não funcionais mal definidos ou ignorados;
- Priorização em uma única dimensão. A priorização baseada exclusivamente em valor de negócio pode trazer problemas relacionados a arquitetura que não foi projetada para acomodar tais requisitos;
- Documentação mínima. Devido ao relacionamento direto com o cliente, muitos times acabam não priorizando a documentação. No entanto, algumas vezes, o time pode precisar dessa documentação, e a falta dela pode causar problemas;
- Falta de verificação dos requisitos. Como não existe um modelo formal detalhado dos requisitos, a verificação de consistências e inspeções formais são raramente realizadas em um contexto ágil;

Além disso, os autores relacionam as práticas e desafios encontrados com alguns riscos previamente mapeados da literatura.

Já o trabalho de Fernández et al. (2016) apresenta o *status quo* e os problemas encontrados no mundo real relacionados a engenharia de requisitos. Os dados foram obtidos através da iniciativa *Naming the Pain in Requirements Engineering* (NaPiRE)<sup>1</sup> que a cada dois anos coleta dados de diversas empresas ao redor do mundo. No artigo de 2016, os autores obtiveram dados de 228 organizações que estão distribuídas em 10 países. Como instrumento de pesquisa foi utilizado um formulário de 33 perguntas divididos em 3 seções: demografia, *status quo* e problemas.

De forma resumida, os problemas citados por Fernández et al. (2016) são:

---

<sup>1</sup><<http://www.mendezfe.org/projects/napire/>>

- Requisitos incompletos ou escondidos;
- Falhas de comunicação entre os o time do projeto e o cliente;
- Mudanças de alvos (objetivos, processos de negócios e/ou requisitos);
- Requisitos mal especificados que estão muito abstratos;
- *Time boxing*/tempo não suficiente geralmente;
- Falhas de comunicação entre os membros do projeto;
- *Stakeholders* com dificuldades em separar requisitos de soluções já conhecidas;
- Suporte do cliente insuficiente;
- Requisitos inconsistentes;
- Fraco acesso as necessidades do usuário e/ou informações de negócio.

Os autores vão além e ainda relacionam as causas e efeitos dos três itens mais citados com os seus possíveis impactos no projeto.

Ao longo dos capítulos 6 e 7, é possível observar que alguns desses problemas apresentados por Ramesh, Cao e Baskerville (2010) e Fernández et al. (2016) serão endereçados ou mencionados pela equipe participante da pesquisa. Por exemplo, essa pesquisa tem como objetivo intervir em problemas relacionados a requisitos incompletos, que podem estar diretamente ou indiretamente ligados a alguns dos desafios mencionados.

### 3 MÉTODO DE DESENVOLVIMENTO DA PESQUISA

Este capítulo apresenta detalhes sobre o método de desenvolvimento da pesquisa. A saber, as seções 3.1, 3.2 e 3.3 apresentam respectivamente, as questões, as premissas e as estratégias de pesquisa.

#### 3.1 QUESTÕES DA PESQUISA

De acordo com Easterbrook et al. (2008), um dos primeiros passos na escolha de um método de pesquisa apropriado é esclarecer as perguntas de pesquisa, pois as mesmas podem direcionar a pesquisa para diferentes estratégias. No entanto, Cunha e Figueiredo (2002) afirmam que em uma pesquisa-ação é importante analisar e refinar as perguntas de pesquisa em cada ciclo. Por isso, a fim de atingir os objetivos previamente definidos na seção 1.3, esse trabalho foi guiado pela seguinte pergunta central de pesquisa:

**RQ:** Como criar requisitos completos em um projeto de desenvolvimento de *software*?

A partir dessa pergunta central, as seguintes questões secundárias também foram definidas:

**RQ1:** Quais são as ferramentas, técnicas ou abordagens que podem ser implementadas em projetos de desenvolvimento de *software* para tornar os requisitos mais completos para o time?

**RQ2:** É possível diminuir os problemas relacionados ao entendimento dos requisitos sem intervenções por parte do cliente?

Estas perguntas de pesquisa direcionaram e ajustaram a forma como a estratégia de pesquisa foi planejada e executada. Permeando assim, por todas as etapas do processo.

#### 3.2 PREMISSAS DA PESQUISA

Durante o planejamento da pesquisa, algumas premissas foram definidas. São elas:

- A organização entendeu que muitos dos seus times tem problemas no entendimento dos requisitos, mas que por limitações de tempo e forma de experimentação, as implementações das ações seriam direcionadas para um projeto em específico.
- Que todo o time do projeto escolhido precisa participar ativamente da implementação, execução e avaliação das ações.

### 3.3 ESTRATÉGIAS DA PESQUISA

Merriam (2009) argumenta que existem diferentes definições para o termo pesquisa, contudo, todas elas possuem em comum a noção de investigar alguma coisa de forma sistemática. Por isso, usando como fundamento os objetivos definidos na seção 1.3 e as perguntas de pesquisa definidas na seção 3.1, foram definidas as estratégias da pesquisa relacionadas a unidade de análise, a metodologia, ao método de abordagem, ao propósito, ao método de pesquisa, e as técnicas de coleta e análise de dados.

#### 3.3.1 Unidade de Análise

Lewis-Beck, Bryman e Liao (2004) definem a unidade de análise como o elemento mais básico de um projeto de pesquisa científico. Para os autores, ele é o objeto de estudo sobre o qual os pesquisadores formarão as suas considerações.

Como já foi mencionado anteriormente, a empresa parceira da universidade é a “dona do problema”. No entanto, para fins de otimizar a pesquisa com todas as suas características experimentais, foi acordado com os gerentes da empresa que mesmo que o problema seja considerado um problema organizacional (diversos projetos na organização apresentavam o mesmo problema), na pesquisa, ele seria tratado no âmbito da equipe de um projeto. Sendo assim, essa equipe é a unidade de análise da pesquisa. Dado que, as ações implementadas e discutidas nos capítulos 5 e 6 serão direcionadas para uma equipe de desenvolvimento de *software*.

#### 3.3.2 Metodologia

Segundo Marconi e Lakatos (2011), “a metodologia qualitativa preocupa-se em analisar e interpretar aspectos mais profundos, descrevendo a complexidade do comportamento humano”. Já a quantitativa, de acordo as autoras, “valem-se de amostras amplas e de informações numéricas”. Como já foi mencionado anteriormente, essa pesquisa visa resolver um problema do mundo real relacionado ao entendimento das pessoas sobre os requisitos de software de um projeto. Por isso, a metodologia qualitativa foi considerada a mais adequada, visto que ela está diretamente associada ao entendimento da experiência em si, e não interessada em prever o número de defeitos gerados por falhas nos requisitos, por exemplo.

No entanto, é importante ressaltar que Gil (2016) afirma que não existem fórmulas prontas para o tipo de pesquisa adotado nesse trabalho, e que muitas vezes, a análise dos dados depende muito da capacidade e do estilo do pesquisador.

#### 3.3.3 Método de Abordagem

De acordo com Marconi e Lakatos (2011), existe uma confusão na literatura entre o termo método e métodos, por isso, as autoras recomendam usar o termo método de abordagem como uma forma de fazer distinção entre os termos. Sendo assim, Bonat (2009), Marconi e

Lakatos (2011) e Gil (2016) os classificam em 4 tipos: indutivo, dedutivo, hipotético-dedutivo e dialético. Os itens abaixo apresentam um resumo da classificação proposta por Bonat (2009).

- **Indutivo:** esse método de abordagem inicia o processo baseado em dados particulares verificados para inferir uma conclusão mais ampla.
- **Dedutivo:** é o oposto do indutivo. Inicia com a generalização e foca no dado particular.
- **Hipotético-dedutivo:** consiste na união dos dois métodos de abordagem anteriormente apresentados. Essa abordagem consiste na escolha de hipóteses que devem ser submetidas a testes para verificar quais resistem às tentativas de refutação e falseamento.
- **Dialético:** esse método apresenta três fases: a tese, a antítese, e a síntese. A ideia defendida pelos pesquisadores é que ele é um método em movimento. O objeto é provado visando a superação da proposta de uma antítese, que irá gerar a síntese, que por sua vez, formará uma nova tese, e assim por diante.

O método indutivo foi o adotado nessa pesquisa, dado que ele parte da observação de fatos cujas causas buscam ser conhecidas. Esse método de abordagem teve grande importância no âmbito das pesquisas sociais, posto que serviu para que estudiosos adotassem a observação como procedimento básico na busca pelo conhecimento científico (GIL, 2016).

### 3.3.4 Propósito

Runeson e Höst (2008) acreditam que diferentes metodologias de pesquisa possibilitam diferentes propósitos. Ou seja, para os autores um tipo de metodologia de pesquisa não pode ser usado como o padrão para todos os propósitos. Por isso, eles diferenciam os propósitos em quatro tipos: exploratório, descritivo, explanatório e de melhoria. E que segundo os autores (RUNESON; HÖST, 2008), eles podem ser resumidos da seguinte forma:

- **Exploratório:** busca entender o que está acontecendo, buscando novas intuições e gerando ideias;
- **Descritivo:** descreve uma situação ou fenômeno;
- **Explanatório:** busca uma explicação para uma situação ou problema;
- **Melhoria:** visa melhorar um determinado aspecto de um fenômeno estudado.

Nesse sentido, essa pesquisa pode ser classificada quanto ao propósito em exploratória e de melhoria, pois visa entender o contexto da equipe, mas também tem como objetivo propor ações que foquem na melhoria do entendimento dos requisitos pelo time.

### 3.3.5 Método de Pesquisa

Para Easterbrook et al. (2008) existem 5 classes de método de pesquisa que são relevantes para a engenharia de *software*. São elas: experimentos controlados, estudos de caso, pesquisas do tipo *survey*, etnografia e pesquisa-ação. Os itens abaixo apresentam um pequeno resumo sobre cada uma dessas classes baseado na descrição proposta por Easterbrook et al. (2008).

- **Experimentos controlados** (incluindo quase-experimentos): é uma investigação de uma hipótese testável onde uma ou mais variáveis são manipuladas com o intuito de medir o efeito gerado em outras variáveis.
- **Estudos de caso**: é uma forma de pesquisa cujo foco visa investigar um fenômeno contemporâneo em seu contexto real mesmo que os limites entre o fenômeno e o contexto não estejam claramente definidos.
- **Pesquisas do tipo *survey***: esse método de pesquisa busca identificar características de uma determinada população. Para esse fim, uma amostra é selecionada e generalizações são feitas com base nas respostas dessa amostra.
- **Etnografia**: é um método de pesquisa que através de observações estuda comunidades de pessoas com o intuito de entender como as relações sociais acontecem.
- **Pesquisa-ação**: nesse método, os pesquisadores procuram resolver um problema do mundo real, mas ao mesmo tempo estudam a experiência de resolver o problema.

O método de pesquisa escolhido para este trabalho foi a pesquisa-ação, em virtude que a presente pesquisa visa estudar uma situação com o propósito explícito de melhorá-la. Mais detalhes sobre esse método de pesquisa foram apresentados na seção 2.2.

### 3.3.6 Técnicas de Coleta e Análise de Dados

Segundo Easterbrook et al. (2008), uma vez que o método de pesquisa foi selecionado, o pesquisador precisa decidir quais as técnicas para a coleta dos dados que serão usadas. No entanto, o autor alerta que é interessante usar múltiplas técnicas para obter os dados de diferentes perspectivas. Sendo assim, a presente pesquisa utilizou as seguintes técnicas: entrevistas semiestruturadas, grupos focais, observações e análise de documentos como forma de obtenção dos dados.

Já para a análise, extração e codificação dos dados, as principais ferramentas usadas foram: Google Docs<sup>1</sup> e MAXQDA<sup>2</sup>. A descrição completa de todas as ferramentas e outros detalhes sobre esse processo serão apresentados no decorrer do capítulo 5.

---

<sup>1</sup><[docs.google.com/document](https://docs.google.com/document)>

<sup>2</sup><[www.maxqda.com](http://www.maxqda.com)>

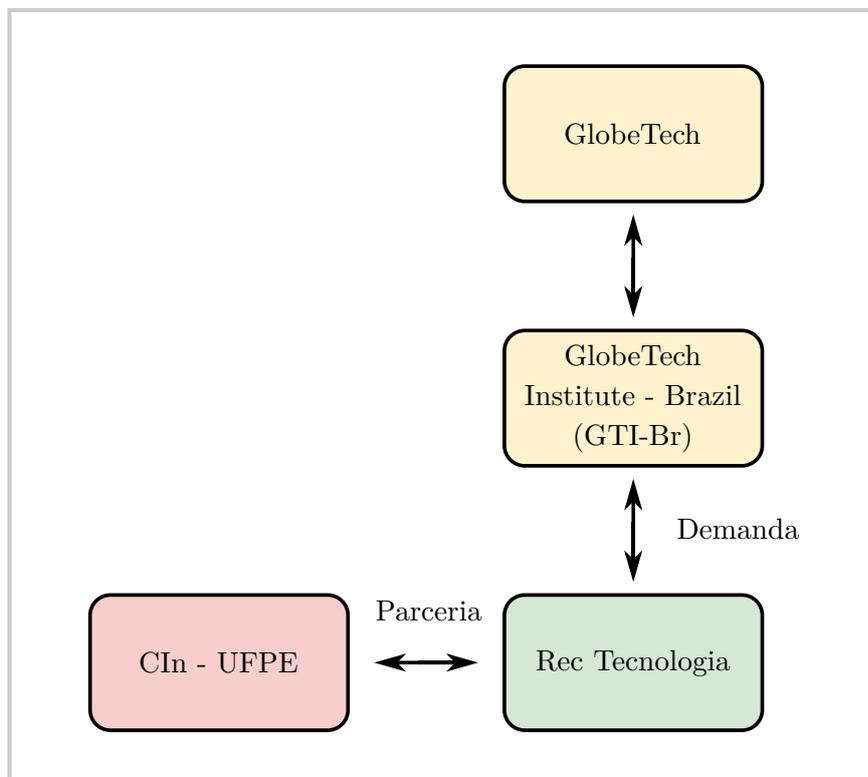
## 4 AMBIENTE DE DESENVOLVIMENTO DA PESQUISA

Encontram-se descritos nesse capítulo o ambiente no qual essa pesquisa foi desenvolvida e as características gerais sobre o projeto indicado para participar da pesquisa. O mesmo irá trabalhar junto com o pesquisador na implementação das ações planejadas.

### 4.1 VISÃO GERAL DOS AMBIENTES DA PESQUISA

A pesquisa foi realizada na Rec Tecnologia, nome fictício para empresa “dona do problema” que por restrições de confidencialidade não pode ter o nome original revelado. A empresa Rec foi criada em 2015 através de uma parceria entre a UFPE e a GlobeTech, uma multinacional que atua no ramo de produtos eletrônicos. No entanto, as equipes da Rec comunicam-se diretamente com um instituto da GlobeTech que fica localizado em outra região do país, o *GlobeTech Institute - Brazil* (GTI-Br). A Figura 3 ilustra como esse relacionamento acontece na prática.

Figura 3 – Relação entre a Rec Tecnologia, UFPE, GTI-Br e *GlobeTech*



**Fonte:** Elaboração Própria

Essa parceria entre a universidade e a multinacional surgiu através da Lei de Informática<sup>1</sup>, uma vez que, a lei concede incentivos fiscais às organizações, através da redução do Imposto

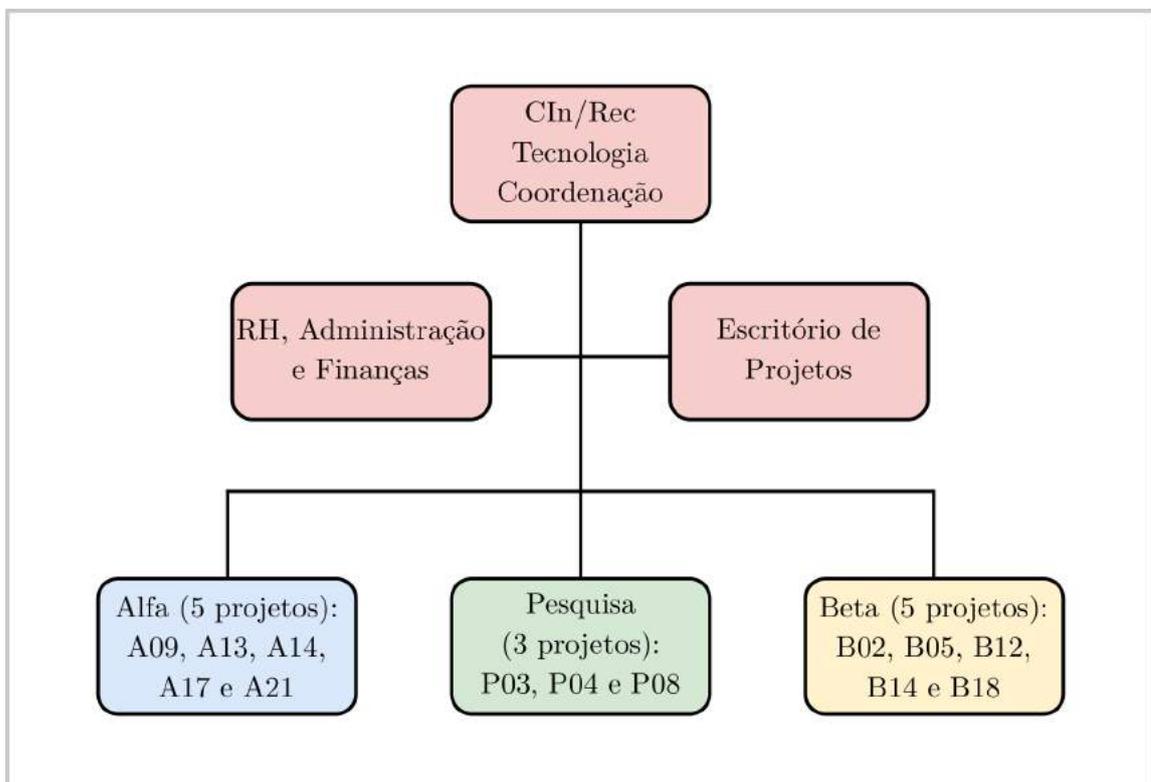
<sup>1</sup><[http://www.planalto.gov.br/ccivil\\_03/LEIS/L8248.htm](http://www.planalto.gov.br/ccivil_03/LEIS/L8248.htm)>

sobre Produtos Industrializados (IPI). Por outro lado, essas empresas precisam investir em projetos de Pesquisa e Desenvolvimento (P&D), e esse convênio entre a Rec Tecnologia e a UFPE surgiu para atingir essa demanda. É importante ressaltar que mesmo sendo uma empresa criada para atender uma demanda de P&D, segundo observações e relatos dos próprios colaboradores, as atividades executadas dentro da Rec são semelhantes as atividades executadas em qualquer empresa de desenvolvimento de *software* da indústria.

#### 4.1.1 Dados demográficos da empresa

Em meados de 2019, a Rec Tecnologia contava com aproximadamente 120 profissionais distribuídos em 3 áreas de atuação (Alfa, Pesquisa e Beta) que estão representadas no organograma apresentado na Figura 4.

Figura 4 – Organograma geral da empresa



Fonte: Elaboração Própria

É importante lembrar que embora as áreas de apoio da organização (Recursos Humanos (RH), Administração e Finanças) estejam agrupadas em apenas uma caixa na Figura 4, as mesmas são departamentos diferentes. O agrupamento na figura acima tem como intuito simplificar a exibição dessas áreas de apoio. Além disso, os colaboradores desses departamentos não participaram ativamente da presente pesquisa. Com exceção de duas pessoas do RH que participaram de uma reunião de apresentação dos resultados da fase de diagnóstico apenas para entender a problemática da pesquisa.

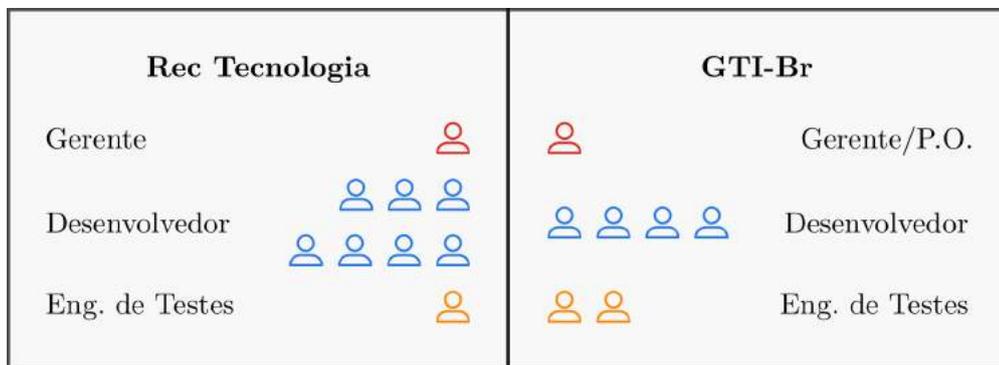
## 4.2 CARACTERIZAÇÃO DO PROJETO

Entre os 13 projetos da Rec apresentados na Figura 4, o projeto B12 foi o escolhido como o mais apropriado para a condução da presente pesquisa durante a apresentação 2a que está disponível na seção F.0.1. Essa decisão foi tomada segundo critérios definidos pelos gerentes envolvidos na pesquisa ao final da etapa de diagnóstico. Outros detalhes sobre essa decisão serão apresentados na subseção 5.3.

A Rec Tecnologia possui 9 colaboradores alocados *full-time* no projeto B12. No entanto, é importante mencionar que existem 7 pessoas no GTI-Br que também integram a equipe do projeto. Segundo os colaboradores da Rec, as pessoas do GTI-Br com perfil técnico (desenvolvedores e engenheiro de testes) que trabalham no dia a dia do projeto, implementando as funcionalidades são vistas como uma extensão da equipe. Já o gerente dessa extensão de equipe, que também fica no GTI-Br, desempenha o papel de *Product Owner* (PO) fazendo a ligação entre o instituto e a Rec Tecnologia. Por isso, o time considera essa pessoa como um representante do cliente.

É importante mencionar que esse representante do cliente é o único canal de comunicação entre o time da empresa “dona do problema” e a GlobeTech. A Figura 5 ilustra como o projeto B12 está estruturado.

Figura 5 – Estrutura do Projeto B12



Fonte: Elaboração Própria

As 14 pessoas envolvidas no projeto B12 com perfil técnico trabalham codificando ou testando uma aplicação desenvolvida para dispositivos móveis e para computadores pessoais. Essa aplicação está em desenvolvimento na Rec Tecnologia desde 2016.

### 4.2.1 Processo de Desenvolvimento

A equipe do projeto B12, assim como todos os outros projetos da Rec Tecnologia são adeptos de algumas práticas ágeis, mas não seguem nenhum *framework* ou método em específico. Como exemplo, pode-se citar que os times seguem um estilo de desenvolvimento iterativo com *sprints* de 15 dias, onde versões são liberados para o cliente, seguindo um cronograma

---

pré-definido. Além disso, o time conduz reuniões diárias de 15 minutos para acompanhar o status da iteração e busca envolver todos do time nesse *mindset* ágil.

Com relação ao processo de desenvolvimento, pode-se dizer que o PO que está no GTI-Br faz a ponte entre a GlobeTech e o time do projeto B12. O mesmo recebe as demandas da GlobeTech, faz um refinamento inicial e repassa ao time. Durante o *planning*, o time divide as funcionalidades por área entre as duas localizações da equipe, Rec Tecnologia e GTI-Br. Como a equipe é distribuída, o *planning* acontece via vídeoconferência com todos os integrantes da equipe.

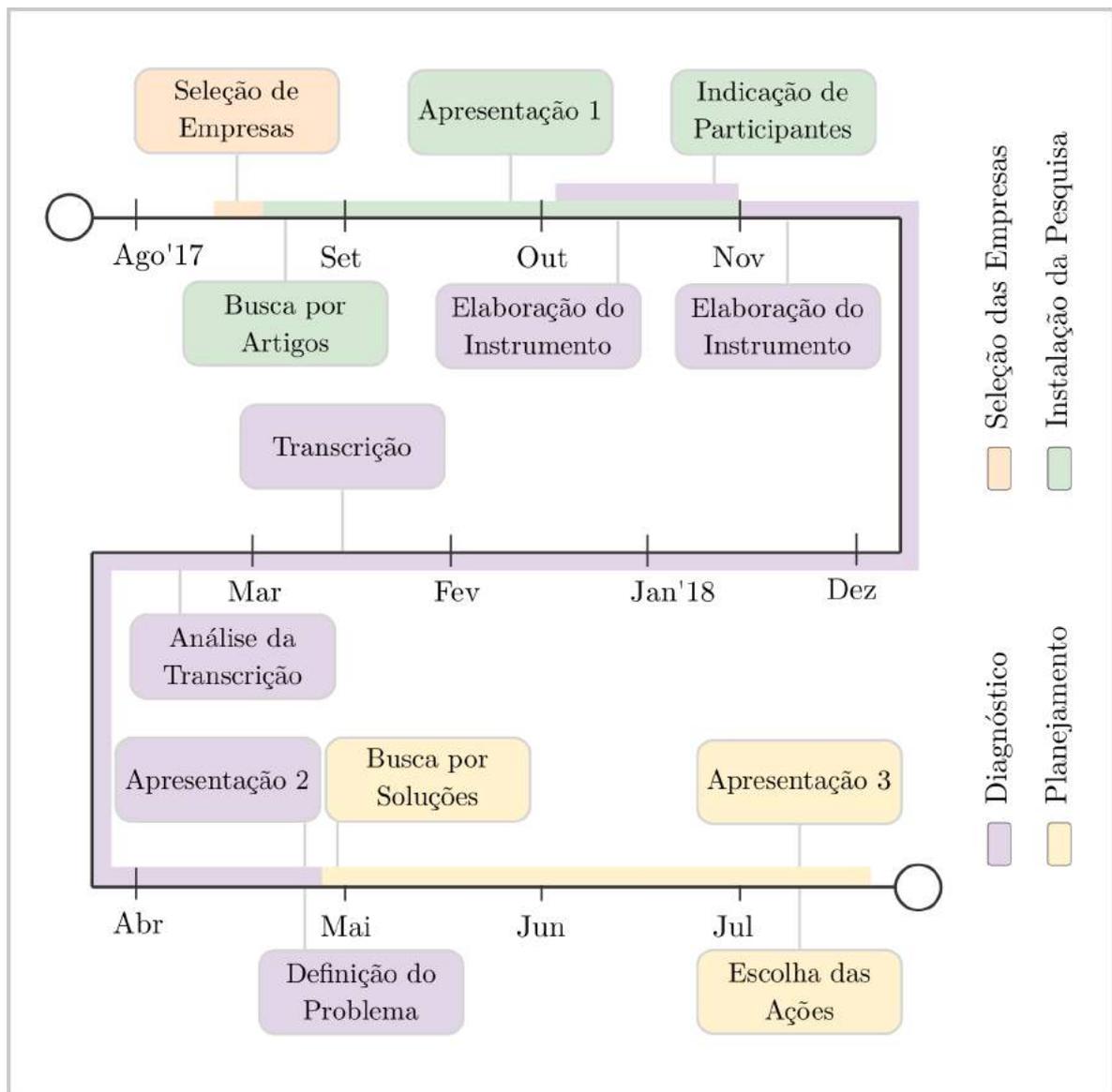
Em seguida, as histórias são divididas entre os 11 desenvolvedores, que precisam finalizá-las antes do período reservado para testes e para o *code freeze*, onde nenhuma modificação deve ser adicionada, apenas correção de *bugs* críticos. Esse processo de validação dos requisitos é feito pelos 3 engenheiros de testes do projeto quando a versão é liberada para os ambientes de testes. Ao final da *sprint*, dependendo do *roadmap* de *releases* da GlobeTech, aquele incremento desenvolvimento durante a iteração pode ir para produção ou não. É importante ressaltar que antes de chegar ao usuário final, a equipe da GlobeTech roda uma nova bateria de testes em cima dessa nova versão.

Em resumo, durante as observações e entrevistas, pode-se notar que embora façam parte de uma equipe só, a parte da equipe que está na Rec Tecnologia e no GTI-Br operam sob culturas organizacionais distintas e isso pode impactar no dia a dia do processo de desenvolvimento do projeto. Por isso, pode-se afirmar que para essa pesquisa, as ações foram planejadas e implementadas em colaboração com a equipe do projeto B12 alocada na Rec Tecnologia apenas.

## 5 DESENVOLVIMENTO DA PESQUISA-AÇÃO

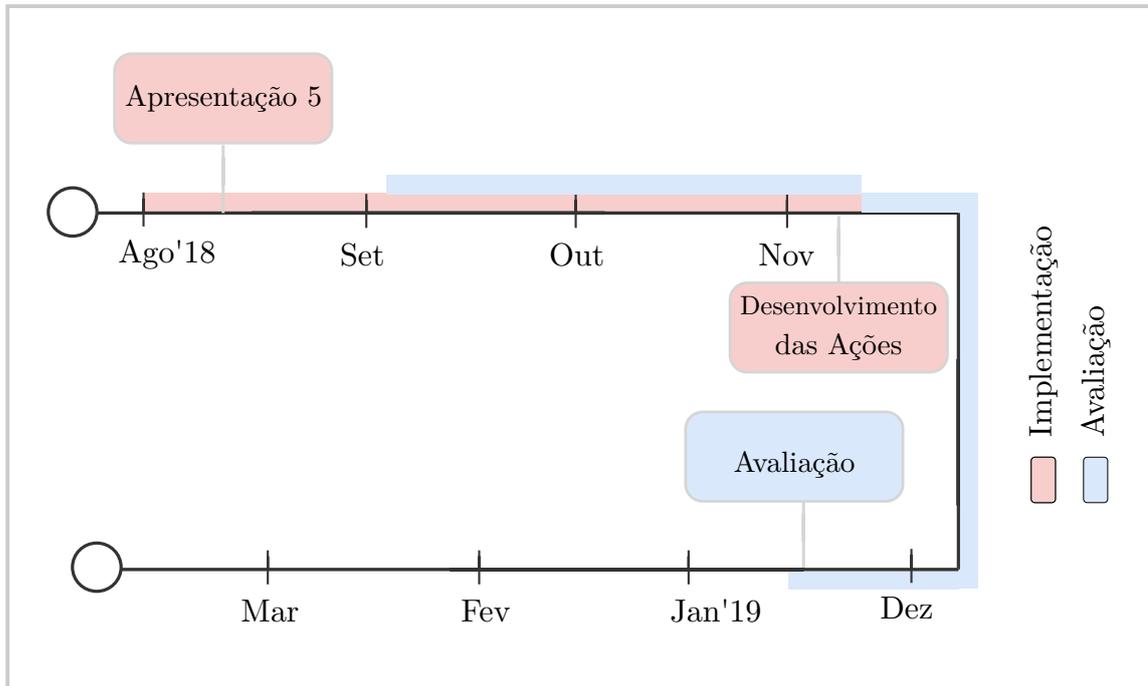
Este capítulo apresenta uma descrição detalhada de todo o processo de desenvolvimento da pesquisa-ação. As Figuras 6 e 7 apresentam uma linha do tempo com as etapas e as principais atividades da pesquisa-ação conduzida na empresa Rec Tecnologia. É importante mencionar que o processo adotado nessa pesquisa e ilustrado nas Figuras 6 e 7 foi baseado no trabalho de Suassuna (2017) e serviu como fundamentação para a elaboração do método que será apresentado no capítulo 6.

Figura 6 – Linha do Tempo com as Etapas e Principais Atividades da Pesquisa-ação - Parte 1



Fonte: Elaboração Própria

Figura 7 – Linha do Tempo com as Etapas e Principais Atividades da Pesquisa-ação - Parte 2



**Fonte:** Elaboração Própria

Vale ressaltar que as atividades posicionadas na linha do tempo ilustradas nas figuras acima podem ser uma atividade pontual resolvida em um dia ou uma atividade contínua que se estende por alguns dias. Por exemplo, as apresentações são atividades pontuais, pois acontecem em um dia e hora previamente marcados. No entanto, a atividade de avaliação, por exemplo, durou um pouco mais de 3 meses. Para esses casos, o posicionamento da caixa, que representa a atividade, está localizado no período de término da mesma.

Além disso, é possível notar que a pesquisa teve duração de aproximadamente 16 meses e foi desenvolvida entre os meses de agosto/2017 a dezembro/2018. No entanto, o cronograma planejado para a presente pesquisa era um pouco mais curto, conforme apresentado logo abaixo na Tabela 1.

Tabela 1 – Cronograma das Etapas da Pesquisa-ação

<b>Atividade</b>	<b>Período Planejado</b>	<b>Período Atual</b>
Instalação da Pesquisa	Ago/2017 a Set/2017	Ago/2017 a Out/2017
Diagnóstico	Set/2017 a Dez/2017	Set/2017 a Abr/2018
Planejamento	Jan/2017 a Mar/2018	Abr/2018 a Jul/2018
Implementação	Abr/2018 a Mai/2018	Ago/2018 a Nov/2018
Avaliação	Mai/2018 a Jun/2018	Set/2018 a Dez/2018

Como pode ser notado na tabela acima, idealmente, o ciclo da pesquisa-ação deveria ter sido executado entre 10 e 11 meses, para que um segundo ciclo da pesquisa pudesse ser rodado em outro projeto. No entanto, a instalação da pesquisa, a transcrição das entrevistas, e a implementação das ações acabaram estendendo-se um pouco mais, e a segunda rodada acabou sendo comprometida. A execução de um segundo ciclo seria bastante interessante para eliminar uma das ameaças a validade da pesquisa tendo em vista que o resultado da pesquisa-ação é sensível ao contexto. Mais detalhes sobre esse tópico serão discutidos na seção 7.1.

Durante o desenvolvimento da pesquisa algumas ferramentas foram utilizadas para auxiliar a análise e a documentação dos artefatos produzidos pela pesquisa. São elas:

- **Ferramentas de e-mail (Gmail<sup>1</sup>) e calendário (Agenda)<sup>2</sup> do Google:** Usados para marcar reuniões, enviar apresentações, solicitar e receber arquivos relevantes para a pesquisa;
- **Ferramentas do Google para criação de documentos<sup>3</sup>, planilhas<sup>4</sup> e apresentações<sup>5</sup>:** Através desses aplicativos, os documentos criados eram salvos automaticamente na conta do pesquisador e poderiam ser facilmente compartilhados com outras pessoas. O Google Documentos também foi bastante útil na transcrição das entrevistas;
- **Software Loopback<sup>6</sup>:** Usado na transcrição das entrevistas para criação de unidades de áudio virtuais;
- **MAXQDA<sup>7</sup>:** Usado na análise das entrevistas porque facilita a codificação e agrupamento de categorias semelhantes;
- **Aplicativo para celular para gravação de voz:** O aplicativo Samsung Voice Recorder<sup>8</sup> foi usado para gravar as entrevistas e algumas discussões sobre a pesquisa.
- **Itens gerais de papelaria:** Papel, lápis e caneta geralmente eram usados nas reuniões para descrever ou ilustrar algum fato.

De forma resumida, a Tabela 2 apresenta alguns números coletados sobre o uso das ferramentas citadas acima.

---

<sup>1</sup><gmail.com>

<sup>2</sup><calendar.google.com>

<sup>3</sup><docs.google.com/document>

<sup>4</sup><docs.google.com/spreadsheets>

<sup>5</sup><docs.google.com/presentation>

<sup>6</sup><rogueamoeba.com/loopback/>

<sup>7</sup><www.maxqda.com>

<sup>8</sup><play.google.com/store/apps/details?id=com.sec.android.app.voicenote>

Tabela 2 – Números Gerais sobre os Artefatos Gerados Durante a Pesquisa

Descrição	Quantidade
E-mails trocados	248
Reuniões agendadas	18
Horas de áudio coletadas	5,6
Documentos criados (incluindo apresentações e planilhas)	32

As seções seguintes apresentam de forma detalhada as atividades relacionadas a preparação para o início da pesquisa e a descrição de cada uma das fases da pesquisa-ação.

### 5.1 SELEÇÃO DA EMPRESA

A etapa de seleção da empresa participante é composta por apenas uma atividade, onde o pesquisador precisa encontrar uma empresa com um problema do mundo real. Essa atividade está representada no fluxograma ilustrado na Figura 21 disponível no Apêndice C. É importante lembrar que essa atividade está alinhada com a pré-condição básica para a condução de uma pesquisa-ação, que é ter uma empresa “dona do problema”, como já visto na seção 2.2.

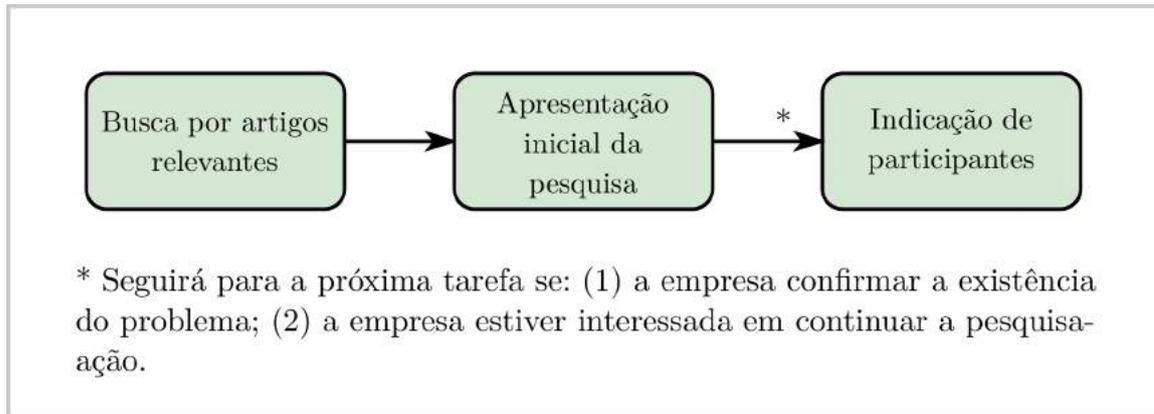
Esse processo de seleção da empresa participante da pesquisa ocorreu de forma bastante simplificada. No caso desse trabalho, o pesquisador não precisou procurar na indústria uma empresa que tivesse interesse em participar da pesquisa. Pelo contrário, a própria empresa através do convênio com a universidade e constantes interações com os professores da instituição apresentou algumas “dores” sobre o problema e se dispôs a cooperar com um projeto de pesquisa.

É importante ressaltar que nessa ocasião, nem a empresa, nem o pesquisador tinham definido especificamente o problema que seria estudado. Nesse momento, o problema apresentado pela empresa era apenas um tema na área de engenharia de *software*. No caso da presente pesquisa, esse tema era relacionado aos requisitos dos projetos.

### 5.2 INSTALAÇÃO DA PESQUISA

A segunda etapa do processo de condução de pesquisa-ação na indústria consiste na fase de instalação da pesquisa. Essa fase é composta por um conjunto de três atividades que serão detalhadas nas subseções seguintes. A Figura 8 apresenta uma visão resumida das atividades dessa etapa.

Figura 8 – Atividades da Etapa de Instalação da Pesquisa



Fonte: Elaboração Própria

É importante mencionar que o fluxograma completo dessa fase está disponível na Figura 21 no Apêndice C.

### Busca por Artigos Relevantes

Pode-se afirmar que a instalação da pesquisa-ação começa uma atividade de preparação, dado que, o pesquisador precisa encontrar na literatura um ou mais artigos relevantes sobre o domínio do problema apresentado pela empresa. Em geral, artigos sobre o estado da arte ou sobre desafios e problemas relacionados ao tema proposto podem ser interessantes.

Dessa forma, a busca por artigos relevantes para esse trabalho foi realizada usando o Google Scholar<sup>9</sup>, e os seguintes termos foram usados na busca: *agile*, *requirements* e *problems*. Depois de analisar alguns artigos, o trabalho de Ramesh, Cao e Baskerville (2010) foi selecionado como um artigo relevante para essa pesquisa porque apresentava práticas e desafios para a engenharia de requisitos em um contexto ágil.

É importante ressaltar que os artigos selecionados nessa atividade servem apenas como uma fundamentação inicial para direcionar e contextualizar o pesquisador na temática do problema apresentado nos primeiros momentos da pesquisa.

### Apresentação Inicial da Pesquisa

Finalizada a seleção dos artigos relevantes, uma apresentação sobre o contexto do problema foi elaborada e apresentada aos gerentes de projeto e coordenadores do grupo de pesquisa, como sugerido na segunda caixa de atividade apresentada na Figura 8.

Essa primeira apresentação está disponível no Apêndice E. Mesmo não estando explícito no documento, durante essa apresentação também foram mencionadas algumas informações relevantes sobre a pesquisa; por exemplo, método de pesquisa, tipo de colaboração esperada entre o pesquisador e a empresa e outros.

<sup>9</sup><scholar.google.com>

---

Ao final da apresentação, os gerentes e coordenadores da empresa Rec Tecnologia asseveraram que, alguns dos problemas apresentados fazem parte do dia a dia dos projetos da empresa, e que eles teriam interesse em prosseguir com a pesquisa.

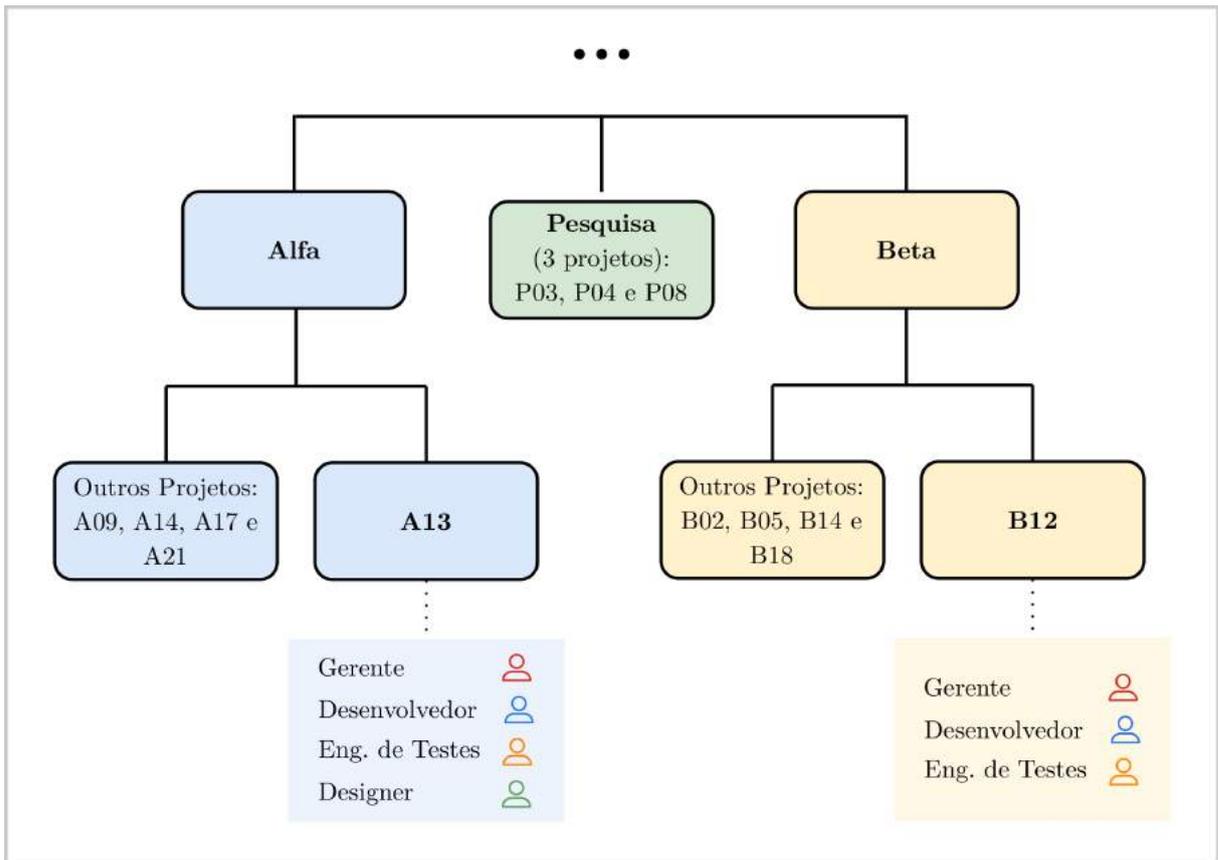
### **Indicação de Participantes**

Geralmente, problemas associados a requisitos estão relacionados ao dia a dia de um projeto. No entanto, de acordo com os colaboradores da Rec Tecnologia presentes na apresentação 1 (disponível no Apêndice E e explicada na subseção anterior), vários projetos já relataram essas mesmas queixas sobre os requisitos das soluções desenvolvidas. Mediante o exposto, foi assumida a premissa que esse problema é inerente a qualquer um dos projetos daquela organização. Dessa forma, logo após a apresentação 1, os coordenadores indicaram dois projetos e seus respectivos gerentes para participarem da fase de diagnóstico.

É importante mencionar que a pesquisa contou com 2 projetos apenas nesse momento inicial, que compreende o final da fase de instalação da pesquisa e toda a fase de diagnóstico. Para as fases seguintes, apenas 1 projeto foi escolhido para prosseguir com a pesquisa. Mais detalhes sobre decisão serão comentados ao final da seção 5.3.

Com o objetivo de obter uma fotografia mais real do cotidiano dos projetos, foi sugerido que as indicações para essa atividade não fossem restritas aos gerentes, mas que incluíssem colaboradores com um perfil mais técnico também. Contudo, essas pessoas provenientes desse outro perfil foram indicadas em um momento futuro pelo coordenador do projeto, como pode ser observado na linha do tempo ilustrada na Figura 6. Ao todo, foram indicados 7 colaboradores (2 gerentes e 5 colaboradores com perfil técnico), cujas posições no organograma da organização são apresentadas na Figura 9.

Figura 9 – Organograma Parcial com Destaque para os Participantes Entrevistados

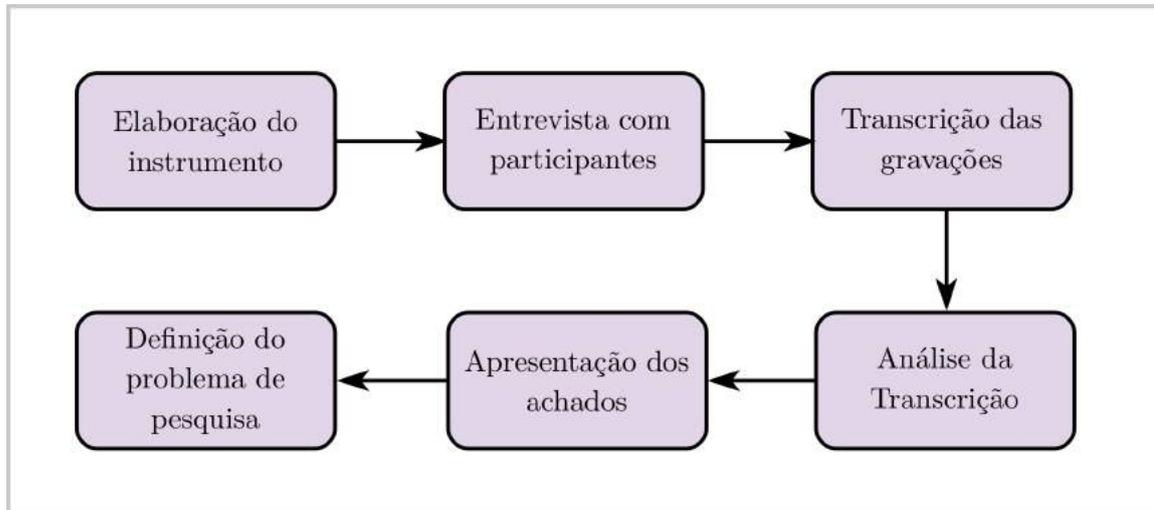


Fonte: Elaboração Própria

### 5.3 DIAGNÓSTICO

A etapa do diagnóstico é composta basicamente por 6 atividades, e a mesma tem como início a construção de um instrumento de pesquisa para o diagnóstico do problema e estende-se até a definição do problema real da pesquisa. A Figura 10 apresenta as principais atividades dessa etapa. É importante lembrar que a etapa abordada nessa seção é a primeira etapa encontrada na literatura para o ciclo de uma pesquisa-ação, como visto na subseção 2.2.3.

Figura 10 – Atividades da Etapa de Diagnóstico



Fonte: Elaboração Própria

### Elaboração do Instrumento

Uma entrevista semiestruturada foi o instrumento de pesquisa escolhido para obtenção de dados nessa fase de diagnóstico do problema. Entrevistas são conhecidas por serem uma técnica direta para coleta de dados, e podem ser usadas para uma variedade de objetivos. Por exemplo, de acordo com Seaman (1999), elas podem ser usadas para obter informações passadas dos entrevistados ou até mesmo para coletar opiniões.

No entanto, segundo Easterbrook et al. (2008), um dos desafios chaves do uso de entrevistas consiste na escolha das perguntas. Por isso, foram usados os seguintes artifícios como forma de mitigar esses possíveis riscos: (1) criação do roteiro de entrevista baseado em entrevistas de estudos já publicados na área; (2) revisão do roteiro por uma segunda pessoa.

A criação do roteiro utilizado nas entrevistas utilizou como base algumas perguntas oriundas dos trabalhos de Ramesh, Cao e Baskerville (2010) e Fernández et al. (2016). Adicionalmente, outras perguntas relacionadas ao processo e a temática do problema também foram adicionadas. Em seguida, o roteiro foi enviado para revisão por uma segunda pessoa atuante na área de engenharia de *software*, mas não participante da pesquisa em busca de *feedbacks* sobre o mesmo.

Ao total, o roteiro elaborado possui 35 perguntas agrupadas em 4 áreas, e está disponível no Apêndice D. No entanto, isso não garante o número de perguntas direcionadas a cada participante, visto que, a entrevista é semiestruturada e novas questões podem aparecer ou serem omitidas a medida que informações vão sendo descobertas.

### Entrevista com Participantes

Como já mencionado anteriormente, sete colaboradores de dois projetos diferentes foram indicados para pesquisa, sendo 2 gerentes, 2 desenvolvedores, 2 engenheiros de teste e 1 *designer*.

Em média, as entrevistas duraram cerca de 27 minutos, e foram realizadas entre outubro e novembro de 2017.

Com o intuito de tornar as entrevistas mais eficazes, o *guideline* proposto por McNamara (2006) foi usado na preparação das entrevistas. Dessa forma, as seguintes recomendações foram empregadas:

- **Ambiente com pouca distração:** Quase todas as entrevistas foram realizadas em uma sala de reunião fechada da Rec Tecnologia. A sala é climatizada e possui boa iluminação. Devido a demanda de outras reuniões na sala, uma das entrevistas foi realizada em uma mesa de trabalho comum da empresa;
- **Explicação sobre o propósito da entrevista:** A entrevista teve início com uma breve apresentação pessoal do pesquisador, e logo em seguida, foram apresentados os seguintes itens: visão geral de uma pesquisa-ação, objetivo da pesquisa e o propósito da entrevista;
- **Explicação sobre o formato e duração prevista para a entrevista:** Os participantes foram informados que a pesquisa tinha um formato semiestruturado e que a qualquer momento, ela poderia ser interrompida caso houvessem dúvidas. Além disso, eles também foram comunicados sobre a duração média das entrevistas;
- **Consentimento para gravar:** Seis dos sete participantes consentiram na gravação dos áudios das entrevistas. Um dos participantes não autorizou a gravação do áudio, mas permitiu que notas fossem registradas, a medida que a entrevista fosse conduzida;
- **Espaço para dúvidas e formas de entrar em contato:** Ao final da entrevista, o pesquisador questionou se os entrevistados lembravam de alguma informação relevante não questionada e disponibilizou um e-mail para contato;

Somente após a finalização dos quatro primeiros itens listados acima que a entrevista era iniciada. É importante lembrar que o roteiro (Apêndice D) usado nas entrevistas foi dividido em áreas que algumas vezes estão ligadas a um perfil específico; por exemplo, perguntas relacionadas a validação dos requisitos poderiam ter sido feitas apenas aos engenheiros de testes; no entanto, algumas dessas perguntas foram feitas a todos os participantes com o objetivo de fazer a triangulação dos dados.

### **Transcrição das Gravações**

A terceira atividade da fase de diagnóstico consiste na transcrição das gravações obtidas nas entrevistas. Essa atividade teve início em meados de novembro de 2017 e só foi finalizada em meados de fevereiro de 2018. Segundo o cronograma inicial, a execução de todas as atividades da fase de diagnóstico deveria levar 3 meses e meio (1,5 semana para elaboração do instrumento, 2 semanas para entrevistas, 2 meses para a transcrição, 2 semanas análise

dos dados e 0,5 semana para apresentação). Ou seja, só essa atividade ocasionou um atraso de um mês no planejamento inicial.

No entanto, essa dificuldade já é conhecida na literatura. Easterbrook et al. (2008) mencionam que o processo de transcrição é um processo lento. Esse processo demanda muita atenção para que o conteúdo da transcrição seja idêntico ao áudio.

Para auxiliar o processo de transcrição, a funcionalidade de “Digitação por Voz” do *Google Docs* foi utilizada, de acordo com o passo a passo abaixo:

1. Os áudios foram adicionados a uma pasta compartilhada do *Google Drive*<sup>10</sup> e baixados em um computador;
2. A entrada de áudio do computador foi alterada para que a execução dos arquivos simulassem a entrada do microfone. Para atingir esse objetivo em um computador com o macOS, a aplicação Loopback foi utilizada;
3. Um documento do *Google Docs* foi criado para cada entrevista;
4. A função “Digitação por Voz” foi acionada, e a execução do áudio era iniciada em sequência;
5. Ao final de cada sentença da transcrição automática, o pesquisador fazia os ajustes pertinentes no texto.

Vale mencionar que essa funcionalidade ajudou o processo de transcrição, mas ela não é eficiente ainda para transcrever a gravação em uma única vez. Durante o processo, percebeu-se que a cada 10 ou 15 segundos, a transcrição automática era finalizada, e o pesquisador precisava identificar no áudio em que momento aconteceu a parada para iniciar o processo novamente. Além disso, o pesquisador precisou conferir todo o texto com o áudio para garantir que o texto estava condizente com a gravação.

A Figura 11 apresenta um trecho de uma das transcrições realizadas. Ao todo, o processo de transcrição gerou 40 páginas de texto.

---

<sup>10</sup><drive.google.com>

Figura 11 – Exemplo de Transcrição do Áudio das Entrevistas

**Designer - Projeto A13**

(...)

**Pesquisador:** Você e o time conseguem entender os requisitos com facilidade?

**Entrevistado:** O pessoal daqui sempre tem problema de entender os requisitos.

**Pesquisador:** Como vocês lidam com isso no dia a dia?

**Entrevistado:** Corriqueiramente pelo menos a sensação é que os requisitos são muito mais críticos, ou eles poderiam ser mais claros. Na minha opinião chega a um ponto que ele não precisava ter uma história de usuário. Poderia ser um requisito normal às vezes atrapalha mais do que facilita.

(...)

**Fonte:** Elaboração Própria

### **Análise da Transcrição**

Ao final da atividade de transcrição (quarta atividade da fase de diagnóstico), é necessário analisar os documentos gerados em busca de achados que podem ser relevantes para a definição do problema.

Por isso, todos os documentos gerados na atividade anterior foram agrupados em dois longos arquivos de texto simples sem formatações (.txt), sendo um para cada projeto. Em seguida, ambos arquivos foram importados no MAXQDA. A decisão de usar uma ferramenta específica para análise qualitativa está relacionada com as facilidades apresentadas pela ferramenta para o processo de codificação aberta.

Basicamente, o processo de análise e síntese dos dados aconteceu da seguinte forma:

1. Os arquivos importados foram lidos previamente pelo pesquisador com o objetivo que o mesmo se tornasse familiar com o texto. Nenhuma codificação foi realizada nesse momento;
2. Em uma segunda leitura, categorias foram adicionadas a trechos relevantes em cada um dos documentos. Um trecho é considerado relevante se o mesmo apresenta alguma oportunidade de melhoria ou algum ponto forte relacionado ao domínio do problema;
3. As categorias semelhantes foram agrupadas;

A Figura 12 representa um exemplo de codificação que foi realizada nessa pesquisa com a transcrição de um dos desenvolvedores do projeto B12.

Figura 12 – Exemplo de Codificação

<b>Requisito incompleto</b>	<p><b>Entrevistado:</b> Assim... digamos olhando para o nosso processo de desenvolvimento ao longo dos anos, eu acho que eu hoje tá mais maduro do que antes. Mas o problema é que o requisito que chega do/da &lt;País sede da multinacional&gt; ele é muito vago. Então, termina que ele aquele acaba sendo... o requisito. Ele chega como uma frase...</p>
-----------------------------	---

**Fonte:** Elaboração Própria

Em seguida, os segmentos codificados foram exportados no formato xlsx (arquivo de planilha do *Microsoft Excel*) para serem usados como *input* para a próxima atividade.

### **Apresentação dos Achados**

Os achados para cada projeto resumidos na etapa anterior foram agrupados em dois grupos chamados de oportunidades de melhoria e pontos fortes, e a apresentação dos mesmos para a Rec Tecnologia foi realizada em dois momentos:

Apresentação 2a: Voltada para os gerentes e coordenadores, mas colaboradores da Rec Tecnologia de áreas meio também participaram; Colaboradores das áreas meio incluem pessoas dos departamentos de RH e Finanças, por exemplo;

Apresentação 2b: Direcionada a todos os colaboradores da empresa. Durante essa apresentação, grande parte dos participantes não conhecia a pesquisa. Por isso, antes da apresentação dos achados, foi realizada uma breve apresentação sobre o pesquisador, objetivos da pesquisa e método de pesquisa adotado.

As apresentações 2a e 2b estão disponíveis nas seções F.0.1 e F.0.2, respectivamente.

### **Definição do Problema de Pesquisa**

No decorrer da atividade anterior, foram apresentadas várias oportunidades de melhorias identificadas nos projetos durante as entrevistas que ocorreram na segunda atividade da etapa de diagnóstico. Durante ambas as apresentações, os colaboradores presentes afirmaram que os itens apresentados eram válidos e que de fato representavam uma fotografia real dos projetos da organização.

No entanto, a presente pesquisa deve acontecer dentro de um espaço de tempo limitado e infelizmente não seria possível atacar todos os itens ao mesmo tempo. Por isso, é necessário que a empresa escolha qual ou quais problemas que ela gostaria de trabalhar.

Depois de muitas discussões, os colaboradores da Rec Tecnologia presentes na apresentação 2a chegaram ao acordo que, os problemas a serem atacados em ordem de prioridade eram requisitos em alto-nível e mudanças de requisitos. Para essa pesquisa, foi considerado o item mais prioritário e ao analisar a literatura, optou-se por usar o termo requisitos incompletos para representar requisitos que não deixam explícitos todas as informações relevantes.

É importante ressaltar que essa decisão foi tomada na apresentação 2a, e foi comunicada a todos os colaboradores da empresa na apresentação 2b. Outro ponto que merece ser mencionado é que as demais oportunidades de melhoria apresentadas que não foram selecionadas podem ser afetadas indiretamente através das ações propostas para o problema selecionado.

### Definição da equipe participante

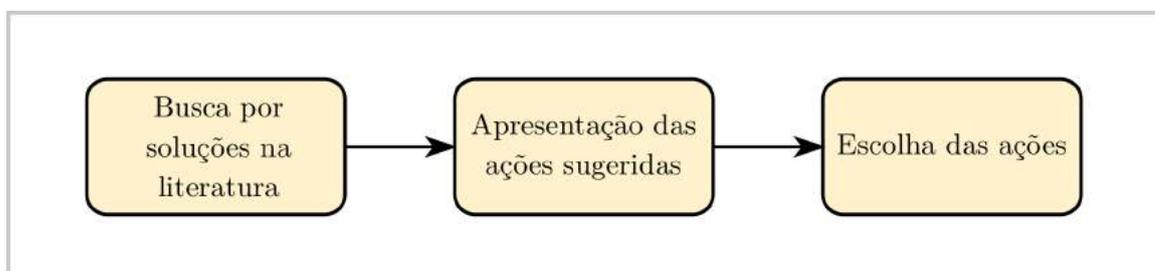
Embora, a pesquisa tenha iniciado a fase de diagnóstico com dois projetos porque em um primeiro momento a empresa afirmou que o problema estudado fazia parte da rotina de todos os projetos da organização, nessa primeira atividade de indicação de participantes, foram indicados 7 colaboradores de 2 projetos quaisquer da organização.

No entanto, finalizada a etapa de diagnóstico, era necessário um projeto real para planejar, implementar e avaliar as ações propostas. Com isso em mente, logo após a definição do problema, os gerentes e colaboradores indicaram um dos projetos que já estava participando da pesquisa, o projeto B12, para dar continuidade com as etapas seguintes da pesquisa. Características gerais desse projeto já foram apresentadas na seção 4.2.

## 5.4 PLANEJAMENTO

Uma vez definido o problema de pesquisa, o pesquisador pode avançar com o *framework* teórico que servirá como fundamentação para a elaboração do plano de ação. A construção dessa base teórica e as demais atividades da etapa de planejamento estão apresentadas na Figura 13.

Figura 13 – Atividades da Etapa de Planejamento



**Fonte:** Elaboração Própria

## Busca por Soluções na Literatura

Como mencionado na subseção 2.2.3, durante a fase de planejamento, as partes envolvidas na pesquisa-ação precisam elaborar um plano contendo ações que serão implementadas. As ações presentes nesse plano podem ser oriundas de alguma teoria já estabelecida ou de achados encontrados na literatura. A Figura 24 disponível no Apêndice C apresenta um fluxograma com mais detalhes sobre essa atividade.

Durante as buscas iniciais para a fundamentação do modelo teórico, não foi encontrada nenhuma teoria relevante que seja direcionada a requisitos incompletos. Por isso, para essa pesquisa, optou-se por usar as soluções apresentadas em diferentes artigos na literatura para a fundamentação do plano de ação. Dessa forma, com o intuito de reunir as soluções que estivessem relacionadas ao problema estudado, uma Revisão Sistemática da Literatura (RSL) foi conduzida baseada na seguinte pergunta de pesquisa “**Q1**: Quais são as ferramentas, técnicas ou abordagens que podem ser implementadas em projetos de desenvolvimento de *software* para tornar os requisitos mais completos para o time?”. Mais detalhes sobre o protocolo de pesquisa usado na revisão da literatura estão disponíveis no Apêndice B. Além disso, as limitações dessa RSL estão listadas na seção 7.1.

Ao final, os artigos selecionados na RSL resultaram na identificação de 23 itens, dos quais 22 estão listados na Tabela 3 abaixo. Esses itens representam as possíveis ações a serem implementadas na etapa seguinte. Para facilitar a descrição, os mesmos foram agrupados em 8 categorias (cliente, cerimônias, elicitação, artefatos, priorização, testes, *release* e outros).

Tabela 3 – Práticas Encontradas na Literatura para Tornar os Requisitos Mais Completos

<b>Categoria</b>	<b>Solução Proposta</b>
Cliente	Cliente <i>on-site</i> ou cliente disponível
	Pessoa com o papel de representante do cliente
	Comunicação face a face com o cliente
	Time com um canal de comunicação direto com o cliente
Cerimônias	Apresentação dos resultados da iteração ( <i>sprint review</i> )
	Reuniões para refinamento de requisitos
Elicitação	Uso de diferentes técnicas para elicitação
Artefatos	Uso de histórias de usuário
	Uso de protótipos
	Uso de <i>Unified Modeling Language</i> (UML)

<b>Categoria</b>	<b>Solução Proposta</b>
	Uso de linguagem natural ou métodos linguísticos
Priorização	Priorização constante
Testes	Adoção de testes de aceitação Adoção de <i>Test-Driven Development</i> (TDD) Escrita de testes para capturar requisitos Rastreabilidade
Release	<i>Releases</i> frequentes <i>Deploy</i> de versões com <i>features</i> não completas
Outros	Proximidade dos membros do time Quebra de requisitos Escopo reduzido ou iterações menores Uso de modelos

É importante mencionar que o *Quality Attribute Workshop* (QAW) proposto por Nord e Tomayko (2006) e que será apresentado como achado da RSL na subseção 6.1.1 não foi incluído nessa lista porque os autores afirmam que o mesmo é apropriado para a primeira iteração de projetos que usam *eXtreme Programming* (XP), que não é o caso da Rec Tecnologia. Por isso, ela não foi considerada uma ação válida para a pesquisa-ação.

### **Apresentação das Ações Sugeridas**

As possíveis ações oriundas da RSL foram agrupadas e apresentadas a todos os membros do projeto B12 em julho/2018. Como essa foi a primeira apresentação exclusiva para a equipe do projeto B12, a sessão iniciou com uma breve apresentação sobre a pesquisa e os seus objetivos. Essa apresentação, também chamada de apresentação 3, levou aproximadamente 1 hora e focou em fornecer uma visão geral sobre todas as ações encontradas na literatura. O documento utilizado durante a apresentação está disponível no Apêndice G.

A medida que os itens estavam sendo apresentados, os colaboradores faziam seus questionamentos e informavam se aquela prática já tinha sido aplicada no projeto ou não. É importante notar que o item “reuniões diárias” está listado na apresentação 3 (Apêndice G), mas não foi citado na subseção acima. O artigo que apresentava uma evidência relacionada a essa possível ação foi removido quando a revisão sistemática foi documentada. No entanto, essa remoção

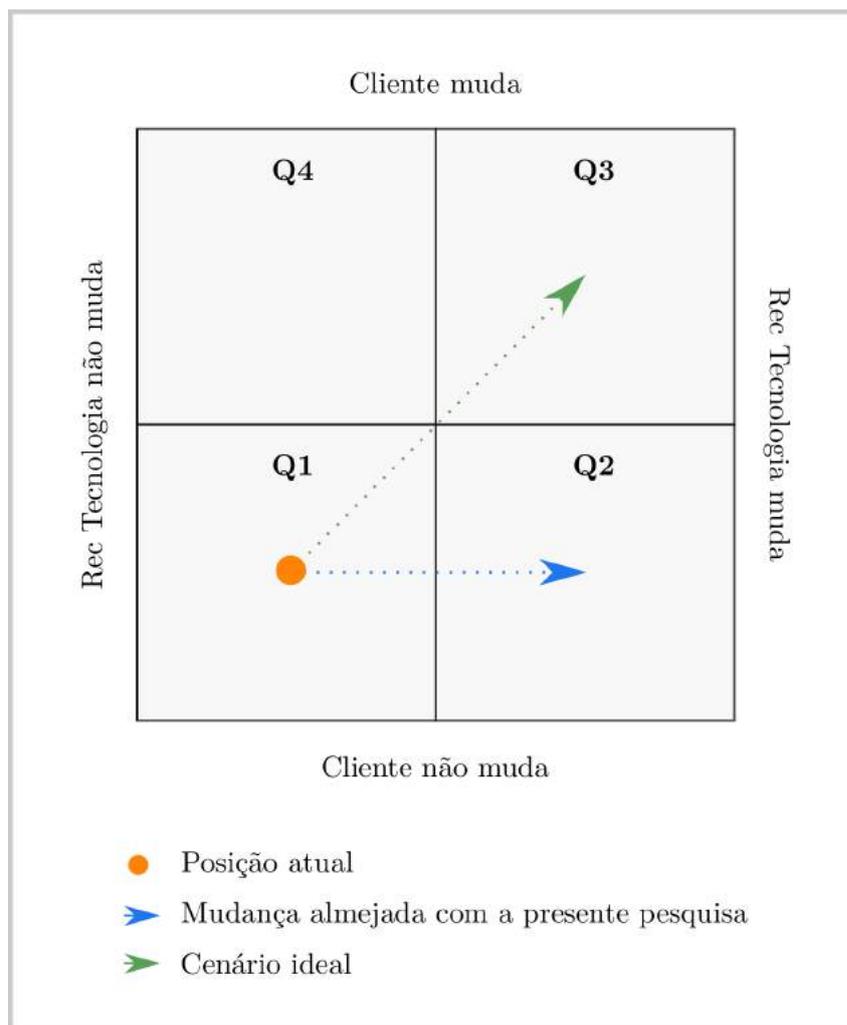
não impactou a pesquisa, visto que o item não havia sido selecionado. Além disso, para a apresentação, alguns itens foram agrupados em um único tópico, mas é possível identifica-los através da descrição dos mesmos (Apêndice G).

### Escolha das Ações

Antes de listar as ações escolhidas pelo projeto B12, é importante discutir a motivação relacionada com a escolha das mesmas.

Poucos dias antes da apresentação mencionada na subseção anterior, o pesquisador, o coordenador de pesquisa da empresa e o professor que acompanhou o trabalho discutiram os próximos passos da pesquisa com relação a implementação das ações encontradas. Essa reunião foi necessária para alinhar a abordagem que seria tomada durante a escolha e o desenvolvimento das ações. Como ilustração, os presentes nessa reunião idealizaram quatro possíveis quadrantes para a pesquisa, como demonstrado na Figura 14.

Figura 14 – Possíveis Quadrantes para a Pesquisa-ação



**Fonte:** Elaborado pelo Professor Orientador da Pesquisa e pelo Coordenador da Rec Tecnologia em uma Reunião de Alinhamento

O primeiro quadrante (Q1) representa a situação atual, onde a Rec Tecnologia apenas aceita e tenta implementar os requisitos incompletos provenientes do GTI-Br e da *GlobeTech*.

Já o Q2, está relacionado com a pergunta de pesquisa “**Q2**: É possível diminuir os problemas relacionados ao entendimento dos requisitos sem intervenções por parte do cliente?”. Nesse caso, o intuito desse quadrante é implementar as mudanças propostas com uma intervenção mínima ou inexistente por parte do cliente. Para esse quadrante, o cliente pode ser participativo nas ações implementadas por terceiros, mas o mesmo não é o responsável pela implementação das mesmas.

O terceiro quadrante representa o cenário ideal onde a empresa “dona do problema” e o instituto que prover as demandas aplicam as mudanças identificadas na pesquisa. Por último, o quadrante Q4 representa o cenário onde ocorre a implementação das ações por parte do GTI-Br, mas nada implementado por parte da Rec Tecnologia. Como o contato dessa pesquisa é direto com a equipe da Rec Tecnologia, o cenário apresentado no Q4 não é uma realidade.

Com o intuito de promover a mudança de forma gradual, o coordenador da pesquisa optou por focar no cenário descrito no Q2, onde as mudanças são implementadas na Rec Tecnologia, mas sem a intervenção do cliente.

Ao final da apresentação discutida na subseção anterior (Apêndice G), os membros do projeto B12 juntamente com o gerente e o coordenador elencaram 6 ações que acharam interessante entre as 21 apresentadas. As ações escolhidas foram:

A1: Reunião de refinamento de requisitos;

A2: *Sprint Review*;

A3: Participação do cliente na escrita ou revisão dos testes de aceitação;

A4: Rastreabilidade entre requisitos e testes;

A5: Escrita de testes para capturar requisitos;

A6: Uso de métodos linguísticos;

No entanto, a sexta ação selecionada acabou não sendo priorizada pela empresa durante o desenvolvimento da pesquisa.

Sobre as ações que não foram selecionados, vale ressaltar que, ações relacionadas a mudanças no envolvimento do cliente não foram selecionadas. Além dessas, algumas ações foram descartadas por que o time já praticava, ou a mesma não era condizente com o contexto da empresa. A Tabela 4 apresenta um resumo da motivação para a não seleção das demais ações propostas.

Tabela 4 – Motivação para a não seleção das ações

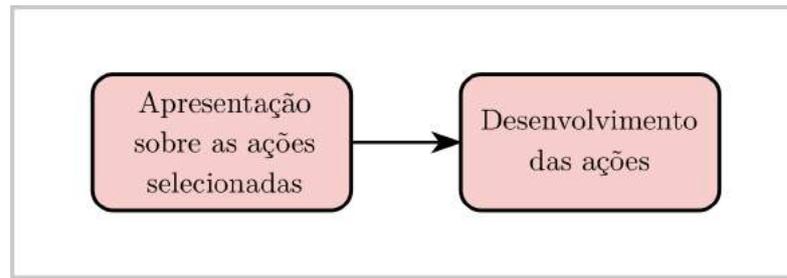
Ação	Motivação
Cliente <i>on-site</i> ou cliente disponível	Como explicado anteriormente, o coordenador de pesquisa da empresa optou por iniciar o processo de mudança sem o envolvimento do cliente.
Pessoa com o papel de representante do cliente	Essa ação já existia parcialmente na equipe, mas a pessoa não era empoderada para tomar decisões. Contudo, seguindo a decisão do não envolvimento ou envolvimento mínimo do cliente, essa ação não foi selecionada.
Comunicação face a face com o cliente	Devido a limitações geográficas, a comunicação entre os times é feita através de áudio-conferências, e tal mudança implicaria em alterações no ambiente do GTI-Br, o que não era desejado no momento.
Time com um canal de comunicação direto com o cliente	Os times já possuíam um grupo em uma determinada ferramenta de comunicação, onde os mesmos poderiam questionar o cliente a qualquer momento.
Uso de diferentes técnicas para elicitação	A elicitação dos requisitos é feita pela <i>GlobeTech</i> em outro continente e a equipe da Rec Tecnologia não tem contato com esse processo. Logo, essa limitação tornava inviável a escolha dessa ação.
Uso de histórias de usuário	O time já utilizava histórias de usuário para descrever os requisitos.
Uso de protótipos	A equipe participante na pesquisa já utilizava versões simplificadas de protótipos para apoiar a descrição dos requisitos.
Uso de UML	O time achou importante a defesa do uso de UML e outros modelos. No entanto, os mesmos acreditam que o uso deve ser pontual para atender determinada necessidade.
Priorização constante	Essa ação não foi escolhida pela equipe devido as restrições previamente definidas.

Ação	Motivação
Adoção de TDD	A escolha dessa ação implicaria em mudanças no processo de desenvolvimento, e a equipe afirmou que ainda não estaria preparada para tal mudança.
<i>Releases</i> frequentes	O <i>roadmap</i> de <i>releases</i> é definido pela <i>GlobeTech</i> . O time da Rec Tecnologia não tem influencia sobre esse processo. Diante disto, optou-se pela não escolha dessa ação.
<i>Deploy</i> de versões com <i>features</i> não completas	Essa ação implicaria em mudanças no processo de <i>releases</i> da <i>GlobeTech</i> . Atualmente, a equipe onde a pesquisa foi desenvolvida não tem visibilidade sobre como o cliente executa esse processo.
Uso de diferentes técnicas para elicitação e documentação	A elicitação dos requisitos é feita pela <i>GlobeTech</i> . Por isso, essa ação não foi escolhida para ser implementada nessa pesquisa.
Proximidade dos membros do time	Os colaboradores já sentavam próximos uns dos outros.
Quebra de requisitos	A equipe do projeto B12 não demonstrou interesse por essa ação. Os requisitos para esse projeto são definidos pela <i>GlobeTech</i> .
Escopo reduzido ou iterações menores	As iterações já tinham duração pre-definida pelo cliente, e a definição do escopo era feita em conjunto com o time da GTI-Br.

## 5.5 IMPLEMENTAÇÃO

Durante a etapa de implementação, o pesquisador e os membros da equipe precisam trabalhar em conjunto na implementação das ações definidas na subseção anterior. Essa etapa consiste em basicamente 2 atividades, conforme ilustrado na Figura 15. Detalhes sobre as atividades são apresentados nas subseções seguintes.

Figura 15 – Atividades da Etapa de Implementação



**Fonte:** Elaboração Própria

### Apresentação das ações selecionadas

Uma vez que as ações a serem implementadas foram escolhidas, uma nova apresentação mais aprofundada em cada uma das ações foi realizada juntamente com os membros do projeto B12. Essa nova apresentação (Apêndice H) tinha como objetivo passar por cada uma das cinco ações escolhidas e priorizadas discutindo os seguintes itens:

- Objetivo da ação;
- Benefícios reportados;
- Boas práticas;
- Exemplos (se possível);
- Referências para outros materiais;

### Desenvolvimento das ações

A implementação dos itens selecionados aconteceu em duas iterações e levou aproximadamente 3 meses. Visto que, entre uma iteração e outra que duram 15 dias cada, o time passou um período sem demanda de novas funcionalidades por parte do cliente. O Quadro 1 apresenta a divisão das atividades por iteração.

Quadro 1 – Divisão das Atividades por Iteração

Iteração 1	Reunião de refinamento de requisitos <i>Sprint review</i>
	Participação do cliente na escrita ou revisão dos testes de aceitação
Iteração 2	Reunião de refinamento de requisitos <i>Sprint review</i>
	Participação do cliente na escrita ou revisão dos testes de aceitação
	Rastreabilidade casos de teste, código e requisitos Escrita de testes para capturar requisitos

É importante mencionar que os termos iteração 1 e iteração 2 indicam apenas as iterações de pesquisa que as ações foram implementadas. Rigorosamente falando, o projeto B12 já tem bem mais *sprints* do que isso, visto que o mesmo está em desenvolvimento na Rec Tecnologia desde 2016.

A divisão das ações em duas iterações foi proposital para que o processo de mudança acontecesse de forma gradual, e os membros da equipe pudessem se familiarizar com a nova rotina. É importante salientar que a implementação dessas ações foram intercaladas com as sessões de avaliação que estas eram focadas em critérios subjetivos e serão detalhadas na seção 5.6.

Os colaboradores da Rec Tecnologia alocados no projeto B12 que ficaram com a responsabilidade de implementar as ações escolhidas. O pesquisador durante essa atividade, serviu apenas como um suporte para o time e acompanhou a implementação das ações.

A primeira ação implementada está associada a realização de reuniões para refinamento de requisitos também chamada pela comunidade ágil de *backlog refinement*. No entanto, a equipe do projeto B12 adotou o termo *pre-planning* para essa cerimônia, e o mesmo será usado em algumas referências a esse evento no decorrer dessa pesquisa.

Durante a primeira iteração de desenvolvimento das ações, o time optou por dividir a segunda ação (apresentação dos resultados da iteração) em duas partes. Uma primeira parte chamada de *sprint review* técnica, cujo foco consiste na transferência de conhecimento entre a equipe da Rec Tecnologia e a extensão da equipe que fica no GTI-Br. Já a segunda parte teria como foco uma apresentação regular dos resultados de uma iteração. No entanto, devido a disponibilidade da GlobeTech, essa segunda parte acabou não sendo implementada.

Sobre a ação relacionada aos testes de aceitação, a GlobeTech não atuou na escrita, mas o PO do GTI-Br participou da revisão dos mesmos. O time da Rec Tecnologia que escreveu os testes de aceitação baseados na descrição das histórias. Muitas vezes, o cliente não tem disponibilidade para escrever os testes de aceitação e isso acaba sendo feito por alguém do próprio time, geralmente um engenheiro de testes. De acordo com Cao e Ramesh (2008), essa iniciativa é válida, desde que o cliente revise e confirme os cenários propostos.

A quinta ação listada na parte inferior do Quadro 1 é referente a matrix de rastreabilidade. Para isso, foi criada uma página no Confluence<sup>11</sup>, ferramenta de colaboração para equipes, onde os requisitos eram relacionados com as suas áreas de impactos, e era possível identificar os testes necessários para cada funcionalidade da aplicação. Esse trabalho é feito manualmente toda *sprint* pelo engenheiro de testes do projeto.

Embora esteja listada por último, a atividade de escrita dos testes para capturar requisitos foi executada logo após a cerimônia de *backlog refinement* pelo engenheiro de testes do projeto. Durante o *planning*, o time já possuía os cenários que seriam explorados durante os testes.

---

<sup>11</sup><[br.atlassian.com/software/confluence](http://br.atlassian.com/software/confluence)>

## 5.6 AVALIAÇÃO

A etapa de avaliação de uma pesquisa-ação consiste na análise dos efeitos teóricos das ações implementadas. Por isso, ao final ou durante a condução de cada iteração apresentada no Quadro 1, atividades relacionadas a verificação dos efeitos da intervenção no problema estudado eram realizadas. Durante todo o processo, avaliações utilizando critérios subjetivos foram conduzidas de três formas: através de grupos focais, observações e entrevistas.

Segundo Morgan (1997), grupos focais são uma forma de entrevista em grupo, que não é resumida a perguntas entre o entrevistador e o entrevistado. Para o autor, o foco nessa sessão consiste na interação do grupo baseado nos tópicos sugeridos pelo pesquisador. Na presente pesquisa, dois grupos focais foram conduzidas com o time do projeto B12 alocados na Rec Tecnologia, sendo uma ao final de cada iteração.

O primeiro grupo focal realizado no final da iteração 1 ressaltou a importância da reunião de refinamento de requisitos e a revisão do cliente nos testes de aceitação. Sobre a reunião de apresentação dos resultados da iteração, o time informou que a ideia da *sprint review* regular foi proposta ao cliente, o mesmo achou-a interessante, mas não teve disponibilidade para participar. Mesmo assim, o time da Rec Tecnologia optou por realizar as duas cerimônias de *review*.

Já durante a etapa 2, foi realizada uma outra sessão de grupo focal, mas também foram feitas observações no dia a dia do projeto e a condução de uma entrevista com um colaborador porque o mesmo estava de férias no dia agendado para o grupo focal 2.

Nas observações, o pesquisador buscou identificar a efetividade das ações implementadas. Por exemplo, o pesquisador participou como observador em uma cerimônia de *planning*, onde a equipe do projeto utilizou a descrição dos cenários em alto nível como complemento a descrição das histórias antes de decidir a estimativa da respectiva história. Sendo assim, os cenários de teste foram úteis para identificar informações que não estavam explícitas nos *Acceptance criteria* (ACs).

O grupo focal 2 aconteceu após a iteração 2 ser finalizada. Para essa sessão, o pesquisador usou uma apresentação (documento disponível no Apêndice I) como roteiro para guiar a sessão. Todas as ações implementadas foram listadas uma por uma, e os colaboradores comentaram suas impressões sobre as mesmas. É importante ressaltar que, semelhantemente ao que aconteceu na primeira iteração, a *sprint review* com o cliente (GlobeTech) não foi implementada.

Para finalizar a etapa de avaliação da pesquisa-ação, uma entrevista com o engenheiro de testes do projeto foi conduzida. A entrevista serviu para corroborar com as percepções relatadas pelos outros colaboradores durante o grupo focal.

Em resumo, as observações e a entrevista realizada com um dos membros da equipe são uma forma de triangular os dados obtidos no grupo focal. Pois, os resultados não foram alterados nas diferentes formas avaliadas. No geral, de acordo com os colaboradores, as ações implementadas tiveram um impacto positivo no alívio do problema inicialmente exposto. Os

detalhes sobre os efeitos dessas ações serão apresentados no capítulo 6.

## 6 RESULTADOS

Este capítulo apresenta os resultados teóricos e empíricos alcançados com a realização da pesquisa-ação. Entre os resultados teóricos, pode-se destacar: uma Revisão Sistemática da Literatura (RSL) e a apresentação do método *Complete Requirements for Software Projects* (CReq-SW), que visa a criação de requisitos completos para equipes de desenvolvimento de *software*. Por outro lado, como resultados empíricos são apresentados, os resultados das ações implementadas e o poder da intervenção do cliente na pesquisa.

### 6.1 RESULTADOS TEÓRICOS DA PESQUISA-AÇÃO

A condução da presente pesquisa gerou como resultado teórico uma RSL que apresenta um conjunto de ações que podem ser selecionadas e implementadas durante a aplicação do método CReq-SW. Ambos são descritos nas subseções seguintes.

#### 6.1.1 Revisão Sistemática da Literatura

Uma Revisão Sistemática da Literatura (RSL) foi conduzida com o intuito de responder a pergunta de pesquisa “**RQ1**: Quais são as ferramentas, técnicas ou abordagens que podem ser implementadas em projetos de desenvolvimento de *software* para tornar os requisitos mais completos para o time?”. Adicionalmente, os achados resultantes dessa RSL foram listados como as possíveis ações que poderiam ser selecionadas pela equipe do projeto B12 durante a etapa de planejamento do método CReq-SW, cuja descrição foi apresentada na seção 5.4.

Após a aplicação dos critérios de inclusão e exclusão (listados na subseção B.0.3), foram selecionados 23 artigos. No entanto, apenas 8 artigos (BJARNASON; WNUK; REGNELL, 2011; CAO; RAMESH, 2008; DAHLSTEDT et al., 2007; LUCIA; QUSEF, 2010; PAETSCH; EBERLEIN; MAURER, 2003; RAMESH; CAO; BASKERVILLE, 2010; SAVOLAINEN; KUUSELA; VILAVAARA, 2010; VLAANDEREN et al., 2011) eram focados em problemas relacionados a requisitos. Os demais artigos apresentaram achados direcionados a como tornar os requisitos mais completos, mas estes são resultados secundários, dado que esse não era o objetivo e o problema apresentado no estudo.

No total, foram encontradas 23 práticas na literatura cujo objetivos são tornar os requisitos mais completos. Ou seja, todas as informações relevantes para um determinado requisito precisam estar contidas nele. Essas práticas foram agrupadas em categorias para simplificar a listagem das mesmas e estão listadas abaixo:

**Cliente:**

- **Cliente *on-site* ou cliente disponível:** Para Ramesh, Cao e Baskerville (2010) e Hanssen e Fægri (2008), o cliente precisa ser colaborativo e está disponível para prover os requisitos certos. Os estudos Huo et al. (2004) e Maurer e

Martel (2002) vão mais além e afirmam que ter um cliente *on-site* ajuda o time a refinar melhor os requisitos.

- **Pessoa com o papel de representante do cliente:** A presença do cliente é fundamental no processo ágil. No entanto, Paetsch, Eberlein e Maurer (2003) afirmam que é comum ter uma pessoa que represente o cliente, mas essa pessoa precisa ser empoderada e capaz de tomar as decisões certas mediante as dúvidas do time. Maurer e Martel (2002) também menciona que esse representante pode ajudar a tornar os requisitos mais claros. Para Strode et al. (2012), alguns desenvolvedores podem fazer esse papel atuando como usuários beta da aplicação.
- **Comunicação face a face com o cliente:** Os autores Ramesh, Cao e Baskerville afirmam que a comunicação face a face com o cliente é a principal fonte para os requisitos (CAO; RAMESH, 2008; RAMESH; CAO; BASKERVILLE, 2010).
- **Time com um canal de comunicação direto com o cliente:** Três artigos defendem que o time precisa ter um canal de comunicação direto com o cliente sem intermediadores de conhecimento para obter requisitos mais claros (PAETSCH; EBERLEIN; MAURER, 2003; CAO; RAMESH, 2008; RAMESH; CAO; BASKERVILLE, 2010).

#### Cerimônias:

- **Apresentação dos resultados da iteração:** Quatro artigos apresentando 3 estudos (PAETSCH; EBERLEIN; MAURER, 2003; CAO; RAMESH, 2008; LEAU et al., 2012; RAMESH; CAO; BASKERVILLE, 2010) mencionam que demonstrações frequentes do produto ao cliente promovem o refinamento dos requisitos através do *feedback*. Essas reuniões para apresentação dos resultados de uma iteração são chamadas pelo Scrum de *sprint review* (SCHWABER; SUTHERLAND, 2017).
- **Reuniões para refinamento de requisitos:** Essa foi a técnica mais citada entre os estudos selecionados (VLAANDEREN et al., 2011; DAHLSTEDT et al., 2007; RAMESH; CAO; BASKERVILLE, 2010; BJARNASON; WNUK; REGNELL, 2011; HANSSON; DITTRICH; ZARNAK, 2006). Os autores desses estudos acreditam que essas sessões servem para checar se os requisitos foram entendidos, discutir os que não foram entendidos e prover detalhes para as funcionalidades que serão implementadas. Os times que usam Scrum e a comunidade ágil geralmente chamam essa ação de refinar os requisitos de *product backlog refinement* ou *grooming* (termo em desuso) (SCHWABER; SUTHERLAND, 2017).
- **Quality Attribute Workshop (QAW):** Para Nord e Tomayko (2006), o QAW é um *workshop* que prover um método explícito para elicitación de requisitos de qualidade. No entanto, o mesmo não foi considerado uma

ação válida na fase de planejamento, como já visto na seção 5.4, porque os autores afirmam que esse tipo de *workshop* é apropriado para a primeira iteração de um projeto que usa XP.

- Elicitação:**
- **Uso de diferentes técnicas para elicitación e documentação:** A aplicação de diferentes técnicas tem como objetivo obter todo o conhecimento possível de todos os envolvidos. Por isso, técnicas como observação e entrevistas são sugeridas por Lucia e Qusef (2010) e Paetsch, Eberlein e Maurer (2003) para elicitación de requisitos.
- Artefatos:**
- **Uso de estória de usuário:** Para Cao e Ramesh (2008) e Bjarnason, Wnuk e Regnell (2011), definir os requisitos usando estória de usuário é uma forma de facilitar a comunicação entre as pessoas de negócio e os engenheiros visando a discussão dos mesmos em detalhes.
  - **Uso de protótipos:** Essa prática é comumente usada quando os requisitos não estão claros. Muitos projetos costumam usar essa prática como uma forma de documentar os requisitos para conversas com o cliente, ao invés de documentá-los formalmente (CAO; RAMESH, 2008; RAMESH; CAO; BASKERVILLE, 2010; FOX; SILLITO; MAURER, 2008; HANSSON; DITTRICH; ZARNAK, 2006).
  - **Uso de UML:** Um dos artigos selecionados (DAHLSTEDT et al., 2007) sugere que o time responsável pela concepção dos requisitos deveria aprender UML para facilitar a comunicação de requisitos.
  - **Uso de linguagem natural ou métodos linguísticos:** Lucia e Qusef (2010) e Dahlstedt et al. (2007) sugerem o uso de linguagem natural ou o uso de métodos linguísticos para elicitación de requisitos.
- Priorização:**
- **Priorização constante:** Em projetos ágeis, os requisitos vão sendo detalhados aos poucos, por isso, que o *backlog* deve ser priorizado constantemente pelo cliente, principalmente antes de cada *sprint* para que o esforço seja direcionado aos itens mais prioritários da lista. O time de desenvolvimento deve participar dessa atividade também, pois o mesmo pode contribuir bastante com *inputs* sobre a saúde do código (CAO; RAMESH, 2008; PETERSEN; WOHLIN, 2010; RAMESH; CAO; BASKERVILLE, 2010).
- Testes:**
- **Adoção de testes de aceitação:** Os artigos de Paetsch, Eberlein e Maurer (2003), Cao e Ramesh (2008), Ramesh, Cao e Baskerville (2010) e Kane et al. (2006) comentam que testes de aceitação podem ser usados como parte da especificação de requisitos. Os artigos também argumentam que idealmente esses artigos deveriam ser escritos pelo cliente, mas caso isso não seja possível, os engenheiros de teste podem escrevê-los e o cliente revisar.

- **Adoção de TDD:** É uma prática bastante popular entre os desenvolvedores de *softwares*, e consiste em escrever os testes antes de codificar a aplicação. Por isso, Cao e Ramesh (2008) e Ambler (2008) acreditam que essa prática promove um forte entendimento dos requisitos.
- **Escrita de testes para capturar requisitos:** Segundo Cao e Ramesh (2008) e Kane et al. (2006), algumas organizações usam testes para capturar de forma completa os requisitos. Por isso, Mishra e Mishra (2011) sugere que junto a descrição dos requisitos, estejam os casos de uso e os cenários já desenvolvidos ao início de cada iteração.
- **Rastreabilidade:** Lucia e Qusef (2010) e Savolainen, Kuusela e Vilavaara (2010) afirmam que informações relacionadas a rastreabilidade entre requisitos, testes e código podem ser úteis para guiar o desenvolvimento das funcionalidades, através da identificação de itens não implementados e na decomposição dos requisitos.

- Release:**
- **Releases frequentes:** Quatro artigos referem-se a entregas constantes como uma forma de obter *feedback* do cliente e assim clarear e refinar os requisitos (LUCIA; QUSEF, 2010; PETERSEN; WOHLIN, 2010; TURK; FRANCE; RUMPE, 2014; LEAU et al., 2012).
  - **Deploy de versões com *features* não completas:** *Releases* pilotos para alguns usuários ou com funcionalidades ainda não finalizadas podem ser consideradas outra forma de contornar os problemas com a completude dos requisitos (CAO; RAMESH, 2008; DAHLSTEDT et al., 2007; RAMESH; CAO; BASKERVILLE, 2010; HANSSON; DITTRICH; ZARNAK, 2006).

Por último, foram listados itens que não pertenciam a nenhuma das outras categorias previamente citadas.

- Outros:**
- **Proximidade dos membros do time:** Segundo Petersen e Wohlin (2010), Ramesh, Cao e Baskerville (2010) e Kane et al. (2006), pessoas envolvidas no processo de desenvolvimento idealmente devem estar ou sentar próximas umas das outras para facilitar as discussões relacionadas as implementações.
  - **Quebra de requisitos:** De acordo com Lucia e Qusef (2010), requisitos complexos devem ser divididos em requisitos menores e mais simples.
  - **Escopo reduzido ou iterações menores:** Os autores Petersen e Wohlin (2009) e Kane et al. (2006) acreditam que iterações menores e escopo reduzido contribuem para requisitos mais detalhados.
  - **Uso de modelos:** Lin (2014) propõe o diagrama *Initial Goal Net* para facilitar o entendimento dos requisitos pelo time. No entanto, Paetsch, Eberlein e Maurer

---

(2003) adverte que qualquer modelo ou diagrama só devem ser desenhados ou usados quando necessário.

É importante mencionar que a descrição da metodologia usada durante a RSL e a lista completa dos artigos selecionados está disponível no Apêndice B. Adicionalmente, as limitações da mesma serão listadas na seção 7.1.

### 6.1.2 O Método

Essa seção apresenta um método para criação de requisitos completos no processo de desenvolvimento de *software*, o CReq-SW (em inglês, *Complete Requirements for Software Projects*). O mesmo teve sua estrutura baseada no trabalho de Suassuna (2017) e o conteúdo foi construído a partir da compreensão dos resultados apresentados pelos seguintes itens:

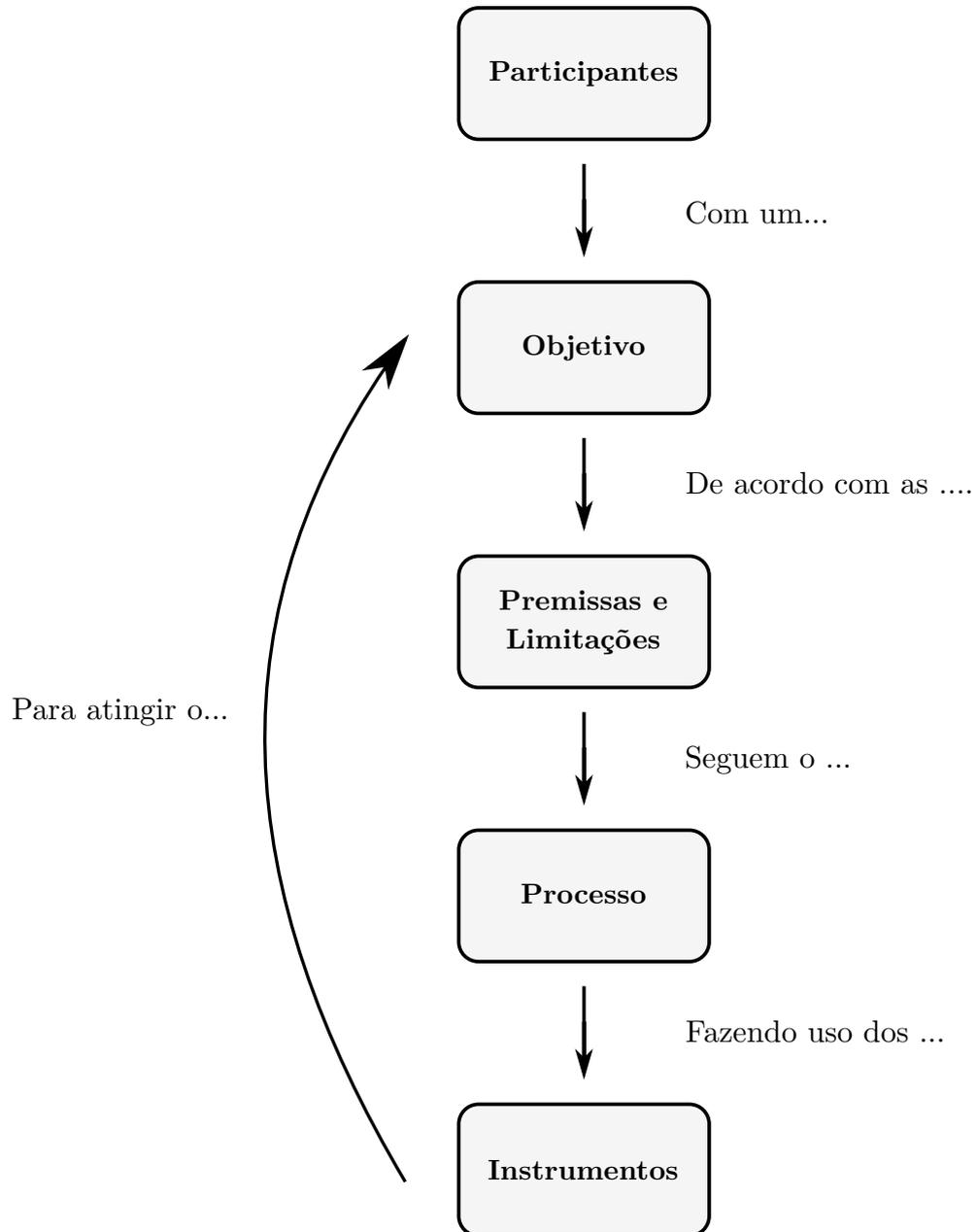
- Resultados obtidos com a Revisão Sistemática da Literatura;
- Processos e instrumentos utilizados durante a condução da pesquisa-ação;
- *Feedbacks* e resultados das avaliações obtidos durante a condução da pesquisa

O método apresentado nessa seção foi desenvolvido em paralelo com a condução da pesquisa-ação, e basicamente o mesmo é composto por cinco componentes principais: objetivo, premissas e limitações, processos, participantes e instrumentos. Os mesmos serão detalhados nas subseções seguinte.

#### 6.1.2.1 Componentes do Método

Com o intuito de facilitar o entendimento e a aplicação do método CReq-SW, a Figura 16 ilustra como os componentes do método estão relacionados.

Figura 16 – Relacionamento entre os Componentes do Método CReq-SW



**Fonte:** Elaboração Própria

As sub-subseções seguintes fornecem uma visão detalhada sobre cada componente do método.

#### **6.1.2.1.1** *Objetivo*

O método CReq-SW apresentado nessa seção tem como objetivo guiar equipes de desenvolvimento, durante o ciclo de vida de um projeto, para que as mesmas possam produzir requisitos de *software* completos.

Durante a aplicação do método, os membros do time de desenvolvimento podem continuamente avaliar a eficácia das ações implementadas, e assim entender quais as ações que produzem impactos positivos no projeto e devem ser mantidas, e quais não geraram os efeitos esperados pela teoria.

#### **6.1.2.1.2 Premissas e Limitações**

Para atingir os objetivos, os participantes da pesquisa precisam ser guiados pelas premissas previamente definidas com a empresa. Por exemplo, os colaboradores precisam estar cientes que a pesquisa-ação é um trabalho colaborativo, e os mesmos são peças fundamentais nesse processo.

Além disso, as limitações (se houver) impostas pela empresa “dona do problema” precisam ser discutidas e levadas em consideração durante toda a aplicação do método.

#### **6.1.2.1.3 Processo**

O processo adotado na adoção do método CReq-SW pode ser dividido em cinco fases ou etapas, onde as mesmas apresentam uma relação quase direta com as etapas do ciclo de pesquisa-ação. Durante o desenvolvimento dessa pesquisa notou-se que existem algumas atividades que fazem parte da pesquisa-ação, mas antecedem as etapas do ciclo formal. Por isso, o ciclo de pesquisa-ação executado nesse trabalho apresenta duas novas etapas que antecedem o ciclo já conhecido, que são as etapas de seleção da empresa e instalação da pesquisa.

Além disso, é importante ressaltar que a etapa de diagnóstico apresentada na seção 5.3 não foi incluída no método. Uma vez que, a implementação do método tem como objetivo trabalhar com requisitos incompletos. Ou seja, o problema já está definido. Mais detalhes sobre todo o ciclo da pesquisa-ação podem ser encontrados na subseção 2.2.3.

A Tabela 5 apresenta a lista completa de atividades agrupadas por fase do processo. Além disso, as Figuras 21, 22 e 23 disponíveis no Apêndice C ilustram todo o processo descrito nessa sub-sub seção. É importante salientar que o processo apresentado nesse trabalho e nas figuras descritas acima não é intrínseco a essa pesquisa. Ou seja, o mesmo pode ser considerado como um guia para condução de qualquer pesquisa-ação com a indústria, desde que sejam feitos os devidos ajustes com relação ao domínio do problema.

Tabela 5 – Descrição do Processo do Método CReq-SW

Etapa do Processo	Atividades
Seleção de Empresa	<ul style="list-style-type: none"> <li>▪ Busca por uma empresa com um problema real relacionado a completude dos requisitos de um projeto de <i>software</i> que esteja disposta a participar da pesquisa;</li> </ul>
Instalação da Pesquisa	<ul style="list-style-type: none"> <li>▪ Busca por artigos relevantes sobre a completude de requisitos. Nesse caso, o trabalho de Kalinowski et al. (2015) pode ser considerado um excelente ponto de partida.</li> <li>▪ Apresentação inicial da pesquisa para a empresa “dona do problema”;</li> <li>▪ Indicação dos colaboradores/projetos participantes na pesquisa;</li> <li>▪ Discussões sobre a confirmação do problema na empresa selecionada e interesse da mesma em prosseguir com a pesquisa;</li> </ul>
Planejamento	<ul style="list-style-type: none"> <li>▪ Leitura dos achados encontrados na RSL apresentados na sub-seção anterior (6.1.1). Adicionalmente, pode ser feita uma busca em outro engenho para obtenção de mais artigos relevantes sobre a temática do problema apresentado;</li> <li>▪ Alinhamento com os gestores sobre o impacto das ações;</li> <li>▪ Apresentação, pelo pesquisador, dessas ações a equipe participante na pesquisa;</li> <li>▪ Definição, pela equipe, dos itens que farão parte do plano de ação;</li> </ul>

Tabela 5 – Descrição do Processo do Método CReq-SW

<b>Etapa do Processo</b>	<b>Atividades</b>
Implementação	<ul style="list-style-type: none"> <li>▪ Apresentação, pelo pesquisador, mais detalhada sobre as ações selecionadas;</li> <li>▪ Implementação, pela equipe, das ações planejadas;</li> </ul>
Avaliação	<ul style="list-style-type: none"> <li>▪ Avaliação, pelo pesquisador, sobre a efetividade das ações implementadas;</li> <li>▪ Discussões com a equipe, sobre a efetividade das ações implementadas;</li> </ul>

#### **6.1.2.1.4 Instrumentos**

Os instrumentos utilizados durante a condução da pesquisa-ação foram desenvolvidos com o intuito de obter informações sobre o problema estudado. A Tabela 6 apresenta a relação de todos os instrumentos utilizados durante a aplicação do método CReq-SW.

Tabela 6 – Instrumentos Utilizados Durante a Aplicação do Método CReq-SW

<b>Instrumento</b>	<b>Descrição</b>
Roteiro para Entrevista	<ul style="list-style-type: none"> <li>▪ Utilizado para levantamento de informações sobre o contexto da empresa;</li> <li>▪ Semi-estruturado com perguntas relacionadas ao processo de desenvolvimento, cliente, projeto e outros. Documento disponível no Apêndice D;</li> </ul>

Tabela 6 – Instrumentos Utilizados Durante a Aplicação do Método CReq-SW

Instrumento	Descrição
Apresentações	<ul style="list-style-type: none"> <li>▪ Utilizadas em quase todas as etapas do processo, as apresentações foram úteis para repassar o embasamento teórico, apresentar achados e guiar as discussões dos grupos focais.</li> <li>▪ Foram realizadas 6 apresentações formais durante a aplicação do método e os arquivos utilizados como guia nas apresentações estão disponíveis nos Apêndices E, F.0.1, F.0.2, G, H e I;</li> </ul>

#### 6.1.2.1.5 Participantes

A condução de uma pesquisa-ação requer um trabalho de colaboração entre o pesquisador e a empresa “dona do problema”. Todavia, para o sucesso da aplicação do método CReq-SW é necessário o envolvimento de outras pessoas. A lista completa dos participantes juntamente com a descrição das suas responsabilidades é apresentada na Tabela 7.

Tabela 7 – Participantes e suas Responsabilidades na Aplicação do Método CReq-SW

Participante	Responsabilidades
Pesquisador	<ul style="list-style-type: none"> <li>▪ Definição e condução das atividades propostas na pesquisa-ação através do embasamento na literatura;</li> <li>▪ Suporte a equipe durante todo o ciclo de desenvolvimento da pesquisa;</li> <li>▪ Participação em reuniões com os coordenadores, gerentes e colaboradores do projeto sobre a pesquisa;</li> <li>▪ Sumarizar e documentar todo o processo e resultados apresentados na pesquisa;</li> </ul>

---

Coordenador e Gerente	<ul style="list-style-type: none"><li>▪ Apresentação dos problemas e contexto da empresa para o pesquisador;</li><li>▪ Seleção e indicação de participantes para participação na pesquisa;</li><li>▪ Definição dos limites da pesquisa na empresa;</li><li>▪ Articulação e participação de reuniões sobre a pesquisa com o pesquisador e os demais envolvidos;</li></ul>
Colaboradores do Projeto	<ul style="list-style-type: none"><li>▪ Desenvolvimento das ações no cotidiano do projeto;</li><li>▪ Participação em reuniões sobre a pesquisa;</li><li>▪ Provisão de <i>feedbacks</i> sobre a pesquisa em desenvolvimento;</li></ul>

---

É importante ressaltar que a tabela apresentada acima está diretamente relacionada ao contexto da empresa “dona do problema” e variações podem ocorrer quando o ambiente de pesquisa for alterado.

## 6.2 RESULTADOS EMPÍRICOS DA PESQUISA-AÇÃO

Durante o ciclo da pesquisa-ação, descrito em detalhes no capítulo 5, foram percebidas mudanças no ambiente de desenvolvimento da pesquisa. Essas mudanças estão relacionadas com a implementação de cinco ações escolhidas pela equipe para tornar os requisitos mais completos. Além disso, nessa seção também é discutido o poder da intervenção do cliente nas ações implementadas.

### 6.2.1 Ações Implementadas

Ao todo, foram cinco ações implementadas pelo projeto B12, como já mencionado anteriormente (seção 5.5) essas ações foram implementadas gradualmente durante duas iterações. As sub-subseções a seguir detalham os achados de cada uma das ações implementadas.

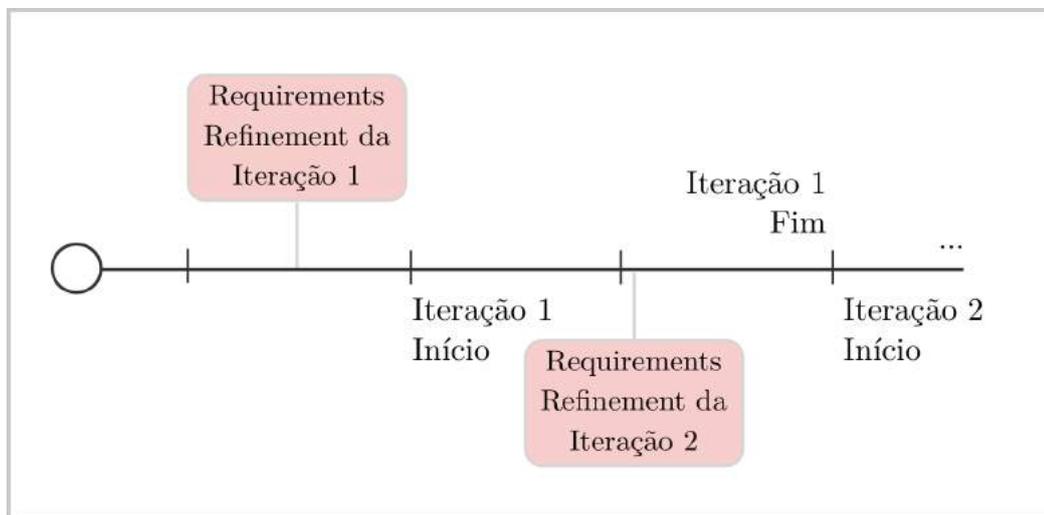
### 6.2.1.1 Reunião de Refinamento de Requisitos (*Backlog Refinement*)

É importante lembrar que durante o ciclo de desenvolvimento do projeto, já apresentado na subseção 4.2.1, os requisitos chegavam ao projeto B12 de forma enxuta e extremamente resumidos. Os colaboradores comentaram que nem todos os fluxos chegavam propriamente ligados, e os mesmos tinham o sentimento que os requisitos não estavam completos. Para eles, era comum faltar informações sobre um ou outro cenário alternativo.

Com isso em mente, as sessões de refinamento de requisitos foram implementadas. A ação de refinar os itens do *backlog* é chamada pelo Scrum de *Product Backlog refinement*. No entanto, a mesma não é listada como um dos eventos oficiais do *framework* (SCHWABER; SUTHERLAND, 2017). O time do projeto B12 chama essa cerimônia de *pre-planning*, e essa foi a primeira ação a ser implementada.

Durante a condução dessa pesquisa com o projeto B12, a reunião de refinamento de requisitos aconteceu duas vezes em iterações distintas. A primeira vez, a cerimônia aconteceu 3 dias antes da iteração 1 e contou com a participação de toda a equipe da Rec Tecnologia e o gestor da GTI-Br. Já durante a segunda rodada, toda a equipe da Rec Tecnologia e da GTI-Br participaram, e a cerimônia aconteceu cinco dias antes do *planning* da iteração 2, como pode ser visto na linha do tempo apresentada na Figura 17.

Figura 17 – Linha do Tempo para a Ação de Refinamento de Requisitos



**Fonte:** Elaboração Própria

De acordo com Schwaber e Sutherland (2017), essas reuniões de refinamento de requisitos tem como objetivo adicionar detalhes, estimativas e ordem aos itens do *backlog*. O time acredita que essa cerimônia é muito importante e provocou mudanças no dia a dia do projeto. Durante os grupos focais, dois colaboradores afirmaram o seguinte quando questionados sobre a importância e efetividade do *pre-planning*:

Desenvolvedor 1: Por a gente (equipe do projeto B12) já ter uma ideia do que iria entrar na *sprint*, a gente já começou a provocar uma conversa

antes e levantar riscos. Então, meio que a gente já envolveu ela (PO) nesses conceitos que a gente iria ter que fechar de cada estória, os riscos e etc...

Desenvolvedor 2: A gente (equipe do projeto B12) realizou isso e foi bem proveitoso pra gente. A gente antecipou dúvidas, e a gente até chegou para reunião com algumas dúvidas sobre algumas *features* já...

Segundo os colaboradores do projeto, além do refinamento dos requisitos discutidos (levantamento de riscos, resolução de dúvidas e criação de novos cenários), essa cerimônia trouxe outros benefícios para o time. Entre eles:

- **Planning mais objetivo:** Um dos colaboradores no grupo focal 2 afirmou que as discussões no *planning* foram mais direcionadas porque o time começou a usar as anotações feitas nas reuniões de refinamento como *input* para as discussões.
- **PO é obrigado a priorizar o backlog:** O gerente do projeto B12 afirmou que quando a reunião é marcada, isso força o PO ou o representante dele a priorizar o *backlog*.
- **Média de estimativas para o planning:** Um colaborador também afirmou que essas sessões ajudam a calcular o esforço total do que será planejado. Segundo ele, a sessão de refinamento ajuda o time a ter uma ideia se determinada funcionalidade vai caber dentro da iteração ou não.
- **Discussões com a participação de todo o time:** Alguns membros relataram também que o time acaba adquirindo mais conhecimento, dado que as discussões são realizadas em conjuntos e qualquer um pode opinar. Sem a sessão de refinamento, os requisitos seriam discutidos apenas entre as pessoas envolvidas no desenvolvimento das estórias, ou seja, um ou dois desenvolvedores.

Após duas iterações com a ação de refinamento de requisitos implementada, o time reconhece que ainda existem alguns problemas e desafios relacionados a essa cerimônia que precisam ser discutidos. Por exemplo, um desenvolvedor do projeto mencionou que o *pre-planning* é efetivo para os itens que estão priorizados no *backlog*, mas isso não garante que todas as estórias adicionadas ao escopo da iteração estejam com os requisitos completos. Segundo ele, já aconteceu uma situação em que no dia do *planning*, o cliente adicionou novas estórias como escopo da iteração que seria iniciada, que não foram refinadas em nenhum *pre-planning*.

Outro item que merece atenção e também foi mencionado nos grupos focais é como o time pode preparar-se para essa reunião ser mais efetiva. Uma sugestão levantada durante a retrospectiva da iteração 2 é que poderia haver uma divisão das estórias entre os membros do time, e a pessoa “dona da estória” ficaria responsável por estudar profundamente os requisitos da mesma e apresentá-los durante a reunião de refinamento. No entanto, essa prática ainda não foi implementada, e é preciso ser cuidadoso com o tempo dedicado a essa atividade de preparação para não atrapalhar o escopo da iteração atual.

### 6.2.1.2 *Sprint Review*

Como já mencionado durante a descrição das atividades da fase de implementação da pesquisa-ação (seção 5.5), a equipe do projeto B12 optou por dividir a *sprint review* em um duas cerimônias, a saber:

- ***Sprint review técnica***: as estórias adicionadas como escopo da *sprint* são divididas entre os membros do time, e eles muitas vezes não possuem uma visão do todo do que foi trabalhado na *sprint*. Por isso, essa reunião foca na transferência de conhecimento entre a equipe da Rec Tecnologia e a sua extensão da equipe que fica no GTI-Br.
- ***Sprint review regular (demo)***: tem como objetivo apresentar o incremento desenvolvido durante a iteração e obter *feedback* do cliente sobre o mesmo. Acontece ao final da *sprint* e conta com a participação do time de desenvolvimento, do PO e outros *stakeholders* do projeto.

O time da Rec Tecnologia achou surpreendente a *sprint review* técnica, um dos desenvolvedores chegou a usar o termo “show de bola”. No entanto, de acordo com o *feedback* obtido nos grupos focais, essa cerimônia foi muito proveitosa para compartilhar conhecimento técnico entre os membros do time. Contudo, não foi encontrada nenhuma evidência que a mesma contribuiu para tornar os requisitos mais completos.

Adicionalmente, devido a restrições de tempo, organização e outros fatores, a *sprint review* regular não foi conduzida com o cliente nem com o seu representante. Em ambas as iterações, o time da Rec Tecnologia propôs a ideia, o representante do cliente achou-a interessante, mas a mesma não foi concretizada. Uma sugestão dada pelo coordenador de pesquisa da Rec Tecnologia foi gravar a cerimônia e em seguida enviar para o cliente ou seu representante, mas essa ideia ainda não foi aplicada.

Em resumo, mediante o exposto, não é possível afirmar que a ação de *sprint review*, seja ela técnica ou regular, contribuiu para tornar os requisitos mais completos.

### 6.2.1.3 Participação do Cliente na Escrita ou Revisão dos Testes de Aceitação

Os testes de aceitação são uma forma de descrever o comportamento da aplicação na forma de um exemplo ou cenário de uso. Por exemplo, o engenheiro de testes do projeto B12 durante uma entrevista no final da iteração 2 forneceu o seguinte exemplo de um teste de aceitação válido do projeto:

**Engenheiro de Testes:** Quando houver um problema de comunicação com o servidor, uma mensagem informando que a aplicação está sem rede deve ser exibida.

Detalhes mais técnicos sobre o cenário descrito acima são adicionadas durante a execução da atividade de levantamento dos cenários de testes, que será descrita na sub-subseção 6.2.1.5.

Conforme o cronograma planejado para a implementação das ações apresentado no Quadro 1, a participação do cliente na escrita ou revisão dos testes aconteceu nas duas iterações. No entanto, no caso do projeto B12, os testes foram escritos pelo engenheiro de testes do projeto, e o representante do cliente atuou apenas como um revisor dos mesmos. Todavia, essa atuação é considerada válida pela literatura como já mencionado anteriormente (seção 5.4). Quando questionado sobre a importância dessa ação durante o primeiro grupo focal, o gerente do projeto fez a seguinte afirmação:

**Gerente do Projeto:** Foi bem importante... Inclusive, dessa reunião, com a revisão do (PO) surgiram novos *improvements* sobre a estória... Isso poderia estourar no ciclo de testes do produto e iria ser aberto como um *bug*.

Através dessas revisões, novo cenários não explícitos anteriormente foram descobertos antes da implementação, o que é bom para o saúde financeira projeto. Uma vez que, o custo para corrigir um defeito durante as fases de requisitos e *design* é bem menor do que o custo para corrigir o mesmo defeito durante os testes ou manutenção. No entanto, essa atividade mesmo não sendo iniciada pelo cliente, requer alguns minutos na sua agenda para a escrita ou revisão dos testes, e a falta de disponibilidade do cliente pode ser um problema nesse caso. Felizmente, isso não aconteceu nessa pesquisa.

Para a equipe, em especial, para o gerente do projeto e para o engenheiro de testes, a validação dos testes de aceitação com o cliente são uma forma de ter os requisitos mais detalhados e o mais importante, confirmados com cliente.

#### 6.2.1.4 Rastreabilidade entre Requisitos e Testes

A rastreabilidade entre requisitos, teste e código foi apontado durante a revisão sistemática (Apêndice B) como uma forma de facilitar a interpretação dos requisitos pelo time. Na fase de implementação das ações, a mesma foi executada na segunda iteração e o processo foi feito manualmente pelo engenheiro de testes. Contudo, no projeto B12, a rastreabilidade restringiu-se a cenário de teste por área impactada e não está diretamente relacionada com o código-fonte da aplicação. A Figura 18 apresenta um exemplo (com dados não reais) da *matrix* criada para o projeto.

Figura 18 – Exemplo da *Matrix* de Rastreabilidade do Projeto B12

	TC01	TC02	TC03	...
Cadastro	X			
Login	X	X		
Logout			X	
...				

**Fonte:** Elaboração Própria

A *matrix* foi criada e é mantida no Confluence, onde todos os membros do time têm acesso e podem consultá-la a qualquer momento. Essa prática foi bastante elogiada pela equipe da Rec Tecnologia e do GTI-Br. No entanto, quando questionado da sua importância para a completude dos requisitos, o engenheiro de testes do projeto fez a seguinte afirmação:

**Engenheiro de Testes:** A *matrix* de me ajuda a entender a dimensão do requisito, a real profundidade das mudanças e onde a gente precisa concentrar a ação. Ela também serve pra ver que um requisito pode ser conflitante com outro, isso já aconteceu...

Embora, tenha apresentado resultados positivos para o projeto, a ação de rastreabilidade entre requisitos, teste e código, que foi implementada através de uma *matrix* entre requisito e testes não forneceu evidências que a mesma contribuiu para que os requisitos do projeto fossem considerados mais completos.

#### 6.2.1.5 Escrita de testes para capturar requisitos

Antes da implementação das ações propostas nessa pesquisa, como já foi descrito na seção 4.2.1, os desenvolvedores do projeto B12 codificavam os requisitos baseados na descrição das histórias. Quando as mesmas eram finalizadas e liberadas para o ambiente de teste, o engenheiro de testes do projeto executava uma série de testes já planejados para aquela história. Na maioria das vezes, os desenvolvedores não investiam tempo pensando sobre cenários alternativos e acabavam não cobrindo determinado fluxo, o que ocasionava os *bugs* que eram abertos pelo testador.

No entanto, depois da implementação da ação discutida nessa sub-sub seção, o processo de levantamento dos cenários de testes passou a ter início logo após a cerimônia de refinamento dos requisitos. Vale lembrar que essa atividade é conduzida pelo engenheiro de testes do projeto, e segundo o mesmo, durante o *pre-planning*, anotações sobre validações que são consideradas relevantes vão sendo feitas para que futuramente possam ser detalhadas. Durante

esse detalhamento, o engenheiro de testes “explode” o teste de aceitação mencionado na sub-seção 6.2.1.3, em vários cenários. Por exemplo, o teste usado como exemplo para a falta de conectividade pode gerar os seguintes cenários:

- Plano de dados ou rede sem fio ativados, mas sem conectividade;
- Plano de dados ou redes sem fio não ativados;
- Sistema operacional do computador não reconheceu o dispositivo de rede;

Nesse momento, os requisitos são analisados profundamente para que todos cenários possíveis sejam levantados. O time chama esses cenários de *Test Outline*, e eles são adicionados em uma seção a parte da descrição da história de usuário no Confluence. Os desenvolvedores do projeto usam esses cenários como guia durante a codificação das histórias. Quando questionado sobre a importância dessa atividade para o entendimento dos requisitos, um desenvolvedor afirmou o seguinte:

Desenvolvedor 3: Ajuda muito. Eles que dizem para mim quando eu terminei porque a gente acha que terminou, mas tem um comportamento x, y, ou z que a gente não tinha previsto. É o conceito de *done* da gente.

De acordo com alguns desenvolvedores e o engenheiro de testes do projeto B12, o número de *bugs* das histórias da *sprint* diminui consideravelmente devido a essa ação. Visto que, segundo eles, quando os cenários levantados já são considerados durante o desenvolvimento, a validação final dos requisitos é muito mais rápida. Além disso, os mesmos ressaltaram a importância do trabalho em parceria entre o desenvolvedor e o testador para entregar um produto de qualidade.

Durante o grupo focal 2, outro desenvolvedor ressaltou ainda a importância desses testes estarem disponíveis já na cerimônia do *planning*, visto que os mesmos podem ajudar o time a ter um melhor entendimento da história e fazer estimativas mais assertivas.

Em resumo, de acordo com as avaliações realizadas, essa ação mostrou-se bastante relevante para encontrar e descrever cenários não explícitos no requisito original. Além disso, a mesma apresentou vários benefícios que vão além da completude dos requisitos.

## 6.2.2 O Poder da Intervenção do Cliente

Durante a atividade de escolha das ações que aconteceu na fase de planejamento da pesquisa-ação (subseção 5.4), foi acordado que o objetivo das ações implementadas era fazer com que o projeto saísse do quadrante Q1 em direção ao Q2, como demonstrado na Figura 14. Por isso, as ações que foram selecionadas e implementadas na Rec Tecnologia visavam iniciar uma mudança com uma intervenção mínima ou até mesmo inexistente por parte do cliente. Essa decisão está diretamente relacionada a pergunta de pesquisa “**RQ2**: É possível diminuir os problemas relacionados ao entendimento dos requisitos sem intervenções por parte do cliente?”

Antes de responder a pergunta **RQ2**, vale salientar que a decisão da não implementação de ações por parte do cliente influenciou todo o processo de escolha das ações. Durante a etapa de planejamento não foram selecionadas nenhuma ação cujo foco da intervenção seria do lado do cliente. Por exemplo, a solução proposta relacionada ao uso de diferentes técnicas para elicitação e documentação dos requisitos não foi considerada no processo de escolha. Isso aconteceu porque o processo de elicitação hoje é feito exclusivamente pela *GlobeTech* com o apoio de seu instituto (GTI-Br) sem a intervenção da Rec Tecnologia. Desta forma, essa possível ação não atendia a decisão recentemente mencionada.

Mediante o exposto, os colaboradores do projeto B12 escolheram cinco ações que eram direcionadas a Rec Tecnologia e que envolviam o cliente de forma passiva. A Tabela 8 apresenta uma visão geral dessas ações com relação a participação do cliente.

Tabela 8 – Ações Escolhidas Relacionadas com a Participação do Cliente

<b>Ação</b>	<b>Participação do Cliente</b>
Reunião de refinamento de requisitos <i>Sprint review</i>	Nenhuma intervenção do cliente é necessária. A <i>GlobeTech</i> e o representante do cliente no GTI-Br deveriam participar da <i>sprint review</i> regular.
Participação do cliente na escrita ou revisão dos testes de aceitação	A <i>GlobeTech</i> ou o representante do cliente no GTI-Br precisam escrever ou revisar os testes de aceitação.
Rastreabilidade casos de teste, código e requisitos	Nenhuma intervenção do cliente é necessária.
Escrita de testes para capturar requisitos	Nenhuma intervenção do cliente é necessária.

É importante mencionar que para a reunião de refinamento de requisitos e a *sprint review* técnica, a equipe do GTI-Br precisa participar, mas a mesma não é considerada como cliente, e sim como uma extensão da equipe. Por isso, que ela não foi listada na tabela acima. Além disso, a Tabela 8 apresenta cinco ações das quais, uma requer a participação do cliente em uma reunião, e uma outra solicita a participação do cliente na escrita ou revisão de testes, que no caso da presente pesquisa, focou-se na revisão dos testes. É importante notar que não houve nenhuma intervenção direta por parte do cliente, o mesmo apenas participou das ações iniciadas pela Rec Tecnologia.

Mesmo com todas as limitações de tempo por parte do cliente ou do seu representante, as

ações foram implementadas com o intuito de promover mudanças relacionadas a completude dos requisitos. É verdade que alguns desafios foram encontrados no meio do caminho, pelo fato do cliente não ser o agente provocador da mudança e exercer um papel fundamental no desenvolvimento de *software*. Como exemplo, pode-se citar a ocasião onde o mesmo adicionou novas estórias ao escopo da *sprint* durante o *planning*. Ou seja, esses itens não passaram pela sessão de refinamento de requisitos como planejado. Outro fato que merece atenção é que existe a dependência do cliente na atualização das prioridades do *backlog* do projeto.

Apesar das limitações mencionadas acima, o time do projeto B12 afirmou que os requisitos do projeto estavam mais “maduros” do que antes da intervenção. Com isso, através das avaliações após a implementação das ações, a presente pesquisa apresentou fundamentos suficientes para responder a pergunta **RQ2** ao afirmar que foi possível provocar mudanças de forma positiva no entendimento dos requisitos sem intervenções por parte do cliente.

## 7 CONCLUSÃO

No capítulo 1 desse trabalho, comentou-se sobre o impacto dos requisitos no sucesso ou falha de um projeto de software. Por isso, atuar nesses aspectos é de suma importância para a academia e para a indústria. Por isso, essa pesquisa foi guiada por um objetivo: criação de um método para propor, analisar e validar ações que podem ser implementadas em projetos de software com o intuito de tornar os requisitos mais completos.

O método proposto, o CReq-SW, possui 5 componentes (objetivo, premissas e limitações, participantes, processo e instrumentos) e foi construído em paralelo com o desenvolvimento dessa pesquisa. É importante mencionar que o mesmo se mostrou eficaz para atingir seu objetivo dentro do contexto dessa pesquisa. Além disso, o processo descrito para o método CReq-SW pode ser adaptado para condução de qualquer pesquisa-ação na indústria.

Como resultados secundários, uma Revisão Sistemática da Literatura (RSL) foi conduzida na etapa de planejamento para identificar as ações que fariam parte do plano de ação da pesquisa. Como resultado, essa RSL gerou 23 ações que podem ser implementadas em projetos de desenvolvimento de software para tornar os requisitos mais completos, e o resultado da mesma também responde a “**RQ1**: Quais são as ferramentas, técnicas ou abordagens que podem ser implementadas em projetos de desenvolvimento de *software* para tornar os requisitos mais completos para o time?”.

Acresce que, das 23 ações resultantes da RSL, 5 foram implementadas. A saber, reunião de refinamento de requisitos, *sprint review*, participação do cliente na escrita ou revisão dos testes de aceitação, rastreabilidade entre requisitos e casos de testes e a escrita de testes para capturar requisitos. Durante duas iterações, o time não conseguiu implementar a *sprint review* regular com o cliente, e a ação relacionada a rastreabilidade ajudou o time a entender o impacto de mudanças, mas não contribuiu para requisitos mais completos. As outras três ações foram consideradas efetivas ao atingirem o objetivo proposto.

Adicionalmente, mesmo sem a intervenção do cliente, a implementação das ações selecionadas geraram uma mudança positiva em relação ao problema estudado. Essa informação é importante para responder a “**RQ2**: É possível diminuir os problemas relacionados ao entendimento dos requisitos sem intervenções por parte do cliente?”.

Um resultado inesperado que foi obtido durante a condução dessa pesquisa foi a detecção da importância dos engenheiros de testes para a obtenção de requisitos mais completos. Três das cinco ações planejadas eram executadas primariamente pelo engenheiro de testes do projeto, ou o mesmo exercia forte influência sobre a atividade. Vale salientar que esse resultado é válido para a equipe do projeto B12, que possui um engenheiro de testes alocado no projeto. Validações em outros contextos são necessárias para generalização desse achado.

Além disso, durante a condução dessa pesquisa-ação foi possível identificar algumas lições aprendidas que podem ser úteis para futuros pesquisados que desejem utilizar a pesquisa-ação

em seus trabalhos. A saber, esses ensinamentos foram:

- **Tenha um cronograma inicial, mas ajuste-se as mudanças:** a pesquisa-ação requer um trabalho de colaboração entre o pesquisador e a empresa. No entanto, essa conexão com a empresa é feita através de seus colaboradores, e os mesmos podem sair de licença médica ou tirar férias, por exemplo. Além disso, algumas atividades podem exigir mais esforço do que o inicialmente planejado, e isto precisa acordado com a empresa participante da pesquisa.
- **As ações sugeridas precisam contemplar o máximo de achados possíveis na literatura:** Os artigos relevantes para a pesquisa, e os artigos que apresentam as possíveis ações precisam passar por uma análise criteriosa. Além disso, o critério de busca dos mesmos deve ser abrangente, mas ao mesmo tempo eficaz para que os artigos importantes sejam incluídos.
- **O impacto das ações pode sair do contexto da empresa “dona do problema”:** pesquisas-ações são caracterizadas por provocarem mudanças com efeitos reais, e essas mudanças podem afetar terceiros além da empresa participante. Por isso, esses impactos e as limitações da pesquisa precisam ser discutidas.
- **A pesquisa-ação pode ser muito custosa:** A condução desse tipo de pesquisa pode ser custosa para a organização já que vai demandar tempo dos seus colaboradores, mas também requer tempo e dedicação do pesquisador. Caso o mesmo tenha outras ocupações, é preciso refletir sobre o tempo necessário para conduzir a pesquisa antes de inicia-la.

## 7.1 LIMITAÇÕES E AMEAÇAS A VALIDADE DA PESQUISA

Vinson e Singer (2008) comentam sobre a importância dos princípios éticos nas pesquisas empíricas em engenharia de *software*. Para os autores, existem 4 princípios que devem ser seguidos. São eles: consentimento informado, beneficência, confidencialidade e valor científico. Esse último compreende a validade do estudo e a importância do tópico de pesquisa. Para os autores, toda decisão sobre ética deve ser feita considerando esses quatro princípios. No entanto, Easterbrook et al. (2008) argumentam que problemas éticos levantados por métodos empíricos não são muito valorizados na literatura de engenharia de *software*.

No entanto, durante essa pesquisa, o pesquisador e o coordenador de pesquisa da empresa “dona do problema” sempre se preocuparam em informar aos participantes os objetivos e os planos almejados com a pesquisa. Além disso, foi comentado também que os dados obtidos seriam apenas para propósitos acadêmicos, e que o nome real da empresa e seus colaboradores não seria divulgado.

Adicionalmente, seguindo a sugestão de Creswell (2015), as medidas abaixo foram tomadas para validar a acurácia dos resultados da pesquisa mitigando as possíveis ameaças a validade da mesma.

- **Triangulação:** Usada para confirmar e validar os achados encontrados durante a pesquisa. Por exemplo, durante a aplicação do método diferentes técnicas de coleta de dados foram usadas. A saber, entrevistas semiestruturadas, observações e análise de documentos.
- **Revisão do instrumento de pesquisa:** Durante a construção do instrumento de pesquisa usado na etapa de diagnóstico, uma pessoa externa a pesquisa, mas da área de engenharia de *software* revisou o roteiro da entrevista e proveu *feedbacks* sobre a mesma.
- **Confirmação dos resultados:** Durante as etapas de diagnóstico e avaliação, os participantes confirmaram que os resultados faziam sentido e eram verdadeiros de acordo com suas perspectivas.

No entanto, como a pesquisa foi executada em apenas um contexto e não existe controle total sobre os fatores que fazem parte do dia do projeto, pode ser que outros efeitos não listados nesse projeto possam ter influenciado os resultados encontrados. Por exemplo, o conhecimento prévio da equipe sobre a situação atual do projeto e do cliente.

Por fim, uma limitação sobre a RSL que deve ser mencionada é que a mesma teve suas buscas restritas aos 100 primeiros artigos resultantes da busca no *Google Scholar*. Os resultados apresentados serviram para gerar a lista de possíveis ações para a pesquisa, mas não é possível afirmar que a mesma contempla todos os artigos sobre soluções para requisitos incompletos disponíveis na literatura.

## 7.2 TRABALHOS FUTUROS

Em primeiro lugar, futuramente, pretende-se refinar os itens encontrados na Revisão Sistemática da Literatura (RSL) considerando as ações apresentadas no trabalho de Kalinowski et al. (2015). As mesmas não foram consideradas na lista de possíveis ações porque esse artigo não foi retornado na busca realizada no *Google Scholar*. Além disso, seria interessante conduzir uma nova etapa de buscas expandindo o número de resultados para o engenho previamente utilizado, e fazer o uso de outros métodos para condução de uma RSL. Por exemplo, o *guideline* do *snowballing* proposto por Wohlin (2014) pode ser uma ferramenta interessante para o processo de refinamento das ações encontradas.

Acresce que, os processos apresentados para a aplicação do método CReq-SW não são inerentes ao contexto dessa pesquisa. A saber, os mesmos foram oriundos no trabalho de Suassuna (2017) e refinados durante o desenvolvimento dessa pesquisa. Sendo assim, como tarefa futura, pode-se citar o seu aperfeiçoamento para publicação do mesmo como um *framework* para condução de pesquisa-ação na indústria.

Adicionalmente, a replicação dessa pesquisa em outros contextos seria interessante para validar os resultados apresentados no capítulo 6, dado que a pesquisa-ação é sensível ao contexto no qual ela foi aplicada. Para facilitar o processo de replicação e comparação com os resultados atuais, todo o processo de desenvolvimento da pesquisa e seus resultados foram detalhados minuciosamente nos capítulos 5 e 6, respectivamente.

Por fim, após o refinamento e validação do método CReq-SW em outros contextos pretende-se publicar esses resultados no meio acadêmico para que os mesmos possam ser utilizados por outros pesquisadores em seus trabalhos.

## REFERÊNCIAS

- AMBLER, S. W. Agile software development at scale. In: *Balancing Agility and Formalism in Software Engineering*. Springer Berlin Heidelberg, 2008. p. 1–12. Disponível em: <[https://doi.org/10.1007/978-3-540-85279-7\\_1](https://doi.org/10.1007/978-3-540-85279-7_1)>.
- BALAJI, S.; MURUGAIYAN, M. S. Waterfall vs v-model vs agile: A comparative study on sdlc. *International Journal of Information Technology and Business Management*, v. 2, n. 1, 6 2012. ISSN 23040777.
- BASKERVILLE, R. L. Distinguishing action research from participative case studies. *Journal of Systems and Information Technology*, Emerald, v. 1, n. 1, p. 24–43, mar 1997. Disponível em: <<https://doi.org/10.1108/13287269780000733>>.
- BASKERVILLE, R. L. Investigating information systems with action research. *Communications of the Association for Information Systems*, v. 2, 1999.
- BASKERVILLE, R. L.; WOOD-HARPER, A. T. A critical perspective on action research as a method for information systems research. *Journal of Information Technology*, v. 11, n. 3, p. 235–246, Sep 1996. ISSN 1466-4437. Disponível em: <<https://doi.org/10.1080/026839696345289>>.
- BECK, K.; BEEDLE, M.; BENNEKUM, A. van; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. *Manifesto for Agile Software Development*. 2001. Disponível em: <<http://www.agilemanifesto.org/>>.
- BJARNASON, E.; WNUK, K.; REGNELL, B. A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In: *Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11*. ACM Press, 2011. Disponível em: <<https://doi.org/10.1145/2068783.2068786>>.
- BONAT, D. *Metodologia de Pesquisa*. 3. ed. Curitiba: IESDE Brasil S.A., 2009. 132 p. ISBN 978-85-387-2219-9.
- CAO, L.; RAMESH, B. Agile requirements engineering practices: An empirical study. *IEEE Software*, Institute of Electrical and Electronics Engineers (IEEE), v. 25, n. 1, p. 60–67, jan 2008. Disponível em: <<https://doi.org/10.1109/ms.2008.1>>.
- CHECKLAND, P. *Systems Thinking, Systems Practice*. Chichester: John Wiley & Sons, 1981. 330 p. ISBN 9780471279112.
- CRESWELL, J. *Educational research : planning, conducting, and evaluating quantitative and qualitative research*. Boston: Pearson, 2015. ISBN 978-0-13-383153-5.
- CUNHA, P. R. da; FIGUEIREDO, A. D. de. Action-research and critical rationalism: A virtuous marriage. In: *ECIS*. [S.l.: s.n.], 2002.

- CURCIO, K.; NAVARRO, T.; MALUCELLI, A.; REINEHR, S. Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, Elsevier BV, v. 139, p. 32–50, may 2018. Disponível em: <<https://doi.org/10.1016/j.jss.2018.01.036>>.
- DAHLSTEDT, G.; KARLSSON, L.; NATT, J.; PERSSON, A. Requirements engineering challenges in market-driven software development – An interview study with practitioners. v. 49, p. 588–604, 2007.
- DAVISON, R.; MARTINSONS, M. G.; KOCK, N. Principles of canonical action research. *Information Systems Journal*, Wiley, v. 14, n. 1, p. 65–86, jan 2004. Disponível em: <<https://doi.org/10.1111/j.1365-2575.2004.00162.x>>.
- DZIDA, W.; FREITAG, R. Making use of scenarios for validating analysis and design. *IEEE Transactions on Software Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 24, n. 12, p. 1182–1196, 1998. Disponível em: <<https://doi.org/10.1109/32.738346>>.
- EASTERBROOK, S. Requirements engineering. In: JIROTKA, M.; GOGUEN, J. A. (Ed.). San Diego, CA, USA: Academic Press Professional, Inc., 1994. cap. Resolving Requirements Conflicts with Computer-supported Negotiation, p. 41–65. ISBN 0-12-385335-4. Disponível em: <<http://dl.acm.org/citation.cfm?id=177970.189996>>.
- EASTERBROOK, S.; SINGER, J.; STOREY, M.-A.; DAMIAN, D. Selecting empirical methods for software engineering research. In: SHULL, F.; SINGER, J.; SJØBERG, D. I. K. (Ed.). *Guide to Advanced Empirical Software Engineering*. 1. ed. The address of the publisher: Springer-Verlag London, 2008. cap. 11, p. 201–213.
- ELGHARIANI, K.; KAMA, N. Review on agile requirements engineering challenges. In: *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*. IEEE, 2016. Disponível em: <<https://doi.org/10.1109/iccoins.2016.7783267>>.
- FERNÁNDEZ, D. M.; WAGNER, S.; KALINOWSKI, M.; FELDERER, M.; MAFRA, P.; VETRÒ, A.; CONTE, T.; CHRISTIANSSON, M.-T.; GREER, D.; LASSENIUS, C.; MÄNNISTÖ, T.; NAYABI, M.; OIVO, M.; PENZENSTADLER, B.; PFAHL, D.; PRIKLADNICKI, R.; RUHE, G.; SCHEKELMANN, A.; SEN, S.; SPINOLA, R.; TUZCU, A.; VARA, J. L. de la; WIERINGA, R. Naming the pain in requirements engineering. *Empirical Software Engineering*, Springer Nature, v. 22, n. 5, p. 2298–2338, oct 2016. Disponível em: <<https://doi.org/10.1007/s10664-016-9451-7>>.
- FOX, D.; SILLITO, J.; MAURER, F. Agile Methods and User-Centered Design : How These Two Methodologies Are Being Successfully Integrated In Industry. *Agile 2008 Conference*, IEEE, p. 63–72, 2008.
- GAUSE, D. Why context matters—and what can we do about it? *IEEE Software*, Institute of Electrical and Electronics Engineers (IEEE), v. 22, n. 5, p. 13–15, sep 2005. Disponível em: <<https://doi.org/10.1109/ms.2005.143>>.
- GIL, A. C. *Métodos e técnicas de Pesquisa Social*. 6. ed. São Paulo: Atlas, 2016. ISBN 978-85-224-5142-5.
- HANSSSEN, G. K.; FÆGRI, T. E. Process fusion: An industrial case study on agile software product line engineering. v. 81, p. 843–854, 2008.

- HANSSON, C.; DITTRICH, Y.; ZARNAK, S. How agile are industrial software development practices ? v. 79, p. 1295–1311, 2006.
- HEIKKILA, V. T.; DAMIAN, D.; LASSENIUS, C.; PAASIVAARA, M. A mapping study on requirements engineering in agile software development. In: *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*. IEEE, 2015. Disponível em: <<https://doi.org/10.1109/seaa.2015.70>>.
- HULL, E.; JACKSON, K.; DICK, J. *Requirements Engineering*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2004. ISBN 1852338792, 9781852338794.
- HUO, M.; VERNER, J.; ZHU, L.; BABAR, M. A. Software Quality and Agile Methods. *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, IEEE, p. 520–525 vol.1, 2004.
- IEEE. IEEE standard glossary of software engineering terminology. *IEEE Std 610.12-1990*, Dec 1990.
- IEEE. IEEE recommended practice for software requirements specifications. *IEEE Std 830-1998*, Oct 1998.
- INAYAT, I.; SALIM, S. S.; MARCZAK, S.; DANEVA, M.; SHAMSHIRBAND, S. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, Elsevier BV, v. 51, p. 915–929, oct 2015. Disponível em: <<https://doi.org/10.1016/j.chb.2014.10.046>>.
- KALINOWSKI, M.; FELDERER, M.; CONTE, T.; SPÍNOLA, R.; PRIKLADNICKI, R.; WINKLER, D.; FERNÁNDEZ, D. M.; WAGNER, S. Preventing incomplete/hidden requirements: Reflections on survey data from austria and brazil. In: *Lecture Notes in Business Information Processing*. Springer International Publishing, 2015. p. 63–78. Disponível em: <[https://doi.org/10.1007/978-3-319-27033-3\\_5](https://doi.org/10.1007/978-3-319-27033-3_5)>.
- KANE, D. W.; HOHMAN, M. M.; CERAMI, E. G.; MCCORMICK, M. W.; KUHLMMAN, K. F.; BYRD, J. A. Agile methods in biomedical software development : a multi-site experience report. v. 12, p. 1–12, 2006.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing Systematic Literature reviews in Software Engineering. *Engineering*, v. 45, n. 4ve, p. 1051, 2007. ISSN 00010782. Disponível em: <<http://www.dur.ac.uk/ebse/resources/Systematic-reviews-5-8.pdf>>.
- KOTONYA, G.; SOMMERVILLE, I. *Requirements engineering : processes and techniques*. Chichester New York: J. Wiley, 1998. ISBN 0471972088.
- KOVITZ, B. L. *Practical Software Requirements: A Manual of Content and Style*. Greenwich, CT, USA: Manning Publications Co., 1998. ISBN 1-884777-59-7.
- KUKHNAVETS, P. Blog, *Agile vs Waterfall: Pros and Cons, Differences and Similarities*. 2016. <<https://blog.ganttpro.com/en/waterfall-vs-agile-with-advantages-and-disadvantages//>>.
- LEAU, Y.; LOO, W. K.; THAM, W. Y.; TAN, S. F. Software Development Life Cycle AGILE vs Traditional Approaches. v. 37, n. Icint, p. 162–167, 2012.

LEWIN, K. Frontiers in group dynamics: li. channels of group life; social planning and action research. *Human Relations*, v. 1, n. 2, p. 143–153, 1947. Disponível em: <<https://doi.org/10.1177/001872674700100201>>.

LEWIS-BECK, M.; BRYMAN, A.; LIAO, T. F. *The SAGE Encyclopedia of Social Science Research Methods*. Sage Publications, Inc., 2004. Disponível em: <<https://doi.org/10.4135/9781412950589>>.

LIN, J. Using Goal Net to Model User Stories in Agile Software Development. *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, p. 1–6, 2014.

LUCIA, A. D.; QUSEF, A. Requirements Engineering in Agile Software Development. v. 2, n. 3, p. 212–220, 2010.

MARCONI, M. de A.; LAKATOS, E. M. *Metodologia Científica*. 6. ed. São Paulo: Atlas, 2011. ISBN 978-85-224-6625-4.

MAURER, F.; MARTEL, S. Extreme programming. rapid development for web-based applications. *IEEE Internet Computing*, v. 6, n. 1, p. 86–90, Jan 2002. ISSN 1089-7801.

MCNAMARA, C. *Field Guide to Consulting and Organizational Development: A Collaborative and Systems Approach to Performance, Change and Learning*. [S.l.]: Authenticity Consulting, LLC, 2006. ISBN 1933719206.

MEDEIROS, J.; ALVES, D.; VASCONCELOS, A.; SILVA, C.; WANDERLEY, E. Requirements engineering in agile projects: A systematic mapping based in evidences of industry. In: *CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering*. Lima, Peru: [s.n.], 2015. p. 460–473.

MERRIAM, S. *Qualitative research : a guide to design and implementation*. San Francisco: Jossey-Bass, 2009. ISBN 978-0-470-28354-8.

MISHRA, D.; MISHRA, A. Complex software project development : agile methods adoption. n. January, p. 549–564, 2011.

MORGAN, D. *Focus groups as qualitative research*. Thousand Oaks, Calif: Sage Publications, 1997. ISBN 0-7619-0342-9.

NORD, R. L.; TOMAYKO, J. E. Software architecture-centric methods and agile development. *IEEE Software*, v. 23, n. 2, p. 47–53, March 2006. ISSN 0740-7459.

NUSEIBEH, B.; EASTERBROOK, S. Requirements engineering: A roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA: ACM, 2000. (ICSE '00), p. 35–46. ISBN 1-58113-253-0. Disponível em: <<http://doi.acm.org/10.1145/336512.336523>>.

PAETSCH, F.; EBERLEIN, A.; MAURER, F. Requirements engineering and agile software development. In: *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003*. [S.l.: s.n.], 2003. p. 308–313. ISSN 1080-1383.

PETERS, M.; ROBINSON, V. The origins and status of action research. *The Journal of Applied Behavioral Science*, SAGE Publications, v. 20, n. 2, p. 113–124, apr 1984. Disponível em: <<https://doi.org/10.1177/002188638402000203>>.

PETERSEN, K.; WOHLIN, C. The Journal of Systems and Software A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *The Journal of Systems & Software*, Elsevier Inc., v. 82, n. 9, p. 1479–1490, 2009. ISSN 0164-1212. Disponível em: <<http://dx.doi.org/10.1016/j.jss.2009.03.036>>.

PETERSEN, K.; WOHLIN, C. The effect of moving from a plan-driven to an incremental software development approach with agile practices An industrial case study. p. 654–693, 2010.

POHL, K. *Requirements engineering : fundamentals, principles, and techniques*. Heidelberg New York: Springer, 2010. ISBN 9783642125775.

PRESSMAN, R. *Software engineering : a practitioner's approach*. New York: McGraw-Hill Higher Education, 2010. ISBN 9780073375977.

RAMESH, B.; CAO, L.; BASKERVILLE, R. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, Wiley, v. 20, n. 5, p. 449–480, sep 2010.

ROBERTSON, S.; ROBERTSON, J. *Mastering the Requirements Process (2Nd Edition)*. [S.l.]: Addison-Wesley Professional, 2006. ISBN 0321419499.

RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, v. 14, n. 2, p. 131, Dec 2008. ISSN 1573-7616. Disponível em: <<https://doi.org/10.1007/s10664-008-9102-8>>.

SAVOLAINEN, J.; KUUSELA, J.; VILAVAARA, A. Transition to agile development - rediscovery of important requirements engineering practices. In: *2010 18th IEEE International Requirements Engineering Conference*. Sydney, Australia: [s.n.], 2010. p. 289–294. ISSN 2332-6441.

SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide*. 2017. <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>>.

SEAMAN, C. B. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on software engineering*, IEEE, v. 25, n. 4, p. 557–572, 1999.

SJOBERG, D. I. K.; DYBA, T.; JORGENSEN, M. The future of empirical methods in software engineering research. In: *Future of Software Engineering (FOSE '07)*. IEEE, 2007. Disponível em: <<https://doi.org/10.1109/fose.2007.30>>.

SOMMERVILLE, I. *Engenharia de software*. Sao Paulo: Pearson Prentice Hall, 2011. ISBN 9788579361081.

SOMMERVILLE, I.; SAWYER, P. *Requirements Engineering: A Good Practice Guide*. [S.l.]: Wiley, 1997. ISBN 0471974447.

STRODE, D. E.; HUFF, S. L.; HOPE, B.; LINK, S. The Journal of Systems and Software Coordination in co-located agile software development projects. *The Journal of Systems & Software*, Elsevier Inc., v. 85, n. 6, p. 1222–1238, 2012. ISSN 0164-1212. Disponível em: <<http://dx.doi.org/10.1016/j.jss.2012.02.017>>.

SUASSUNA, M. *Método para Gerenciamento da Motivação e Satisfação de Engenheiros de Software no Desenvolvimento de Projetos – M3S–SE*. Tese (Doutorado) — Universidade Federal de Pernambuco, Recife, 8 2017.

SUSMAN, G. I. *Action Research: A sociotechnical systems perspective*. London: Sage Publications, 1983. 102 p.

SUSMAN, G. I.; EVERED, R. D. An assessment of the scientific merits of action research. *Administrative Science Quarterly*, [Sage Publications, Inc., Johnson Graduate School of Management, Cornell University], v. 23, n. 4, p. 582–603, 1978. ISSN 00018392. Disponível em: <<http://www.jstor.org/stable/2392581>>.

The Standish Group. *Chaos Report*. 1994. Disponível em: <[https://www.standishgroup.com/sample\\_research\\_files/chaos\\_report\\_1994.pdf](https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf)>.

TRIST, E. L. Engaging with large-scale systems. In: *Experimenting with Organizational Life: The Action Research Approach*. Boston, MA: Springer US, 1976. p. 43–57. ISBN 978-1-4613-4262-5. Disponível em: <[https://doi.org/10.1007/978-1-4613-4262-5\\_5](https://doi.org/10.1007/978-1-4613-4262-5_5)>.

TURK, D.; FRANCE, R. B.; RUMPE, B. Assumptions underlying agile software development processes. *CoRR*, abs/1409.6610, 2014. Disponível em: <<http://arxiv.org/abs/1409.6610>>.

VINSON, N. G.; SINGER, J. A practical guide to ethical research involving humans. In: \_\_\_\_\_. *Guide to Advanced Empirical Software Engineering*. London: Springer London, 2008. p. 229–256. ISBN 978-1-84800-044-5. Disponível em: <[https://doi.org/10.1007/978-1-84800-044-5\\_9](https://doi.org/10.1007/978-1-84800-044-5_9)>.

VLAANDEREN, K.; JANSEN, S.; BRINKKEMPER, S.; JASPERS, E. The agile requirements refinery : Applying SCRUM principles to software product management. *Information and Software Technology*, Elsevier B.V., v. 53, n. 1, p. 58–70, 2011. ISSN 0950-5849. Disponível em: <<http://dx.doi.org/10.1016/j.infsof.2010.08.004>>.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering - EASE '14*. ACM Press, 2014. Disponível em: <<https://doi.org/10.1145/2601248.2601268>>.

ZAVE, P. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 29, n. 4, p. 315–321, dez. 1997. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/267580.267581>>.

## APÊNDICE A – PROBLEMAS COMUNS NA ENGENHARIA DE REQUISITOS

A Tabela 8 apresenta a lista completa dos problemas relacionados a engenharia de requisitos citados por cinco artigos (INAYAT et al., 2015; MEDEIROS et al., 2015; HEIKKILA et al., 2015; ELGHARIANI; KAMA, 2016; CURCIO et al., 2018). É importante ressaltar que a coluna 2 não apresenta todas as perguntas de pesquisa do trabalho em questão, mas sim, a pergunta relacionada ao problema de requisitos.

Tabela 9 – Problemas Comuns na Engenharia de Requisitos em Contextos Ágeis

Artigo	Pergunta de Pesquisa	Problemas citados
Inayat et al. (2015)	Quais são os desafios práticos da engenharia de requisitos ágil?	Documentação mínima Disponibilidade do cliente Estimativas de orçamento e cronograma Arquitetura inapropriada Negligenciar requisitos não funcionais Incapacidade e concordância com o cliente Limitações de contrato e volatilidade dos requisitos Mudanças de requisitos e avaliação de mudança
Medeiros et al. (2015)	Quais são os desafios e limitações das técnicas de engenharia de requisitos em projeto ágeis?	Expectativas do cliente não são atingidas Interação usuário-desenvolvedor inadequada Usuários não sabem o que eles querem Baixa disponibilidade do cliente Documentação insuficiente para implementação, manutenção e treinamentos Compartilhamento de documentação ineficiente Dificuldade em transferir o conhecimento de um projeto para um grande time Validação de requisitos sem a perspectiva do cliente

Tabela 9 – Problemas Comuns na Engenharia de Requisitos em Contextos Ágeis

Artigo	Pergunta de Pesquisa	Problemas citados
Medeiros et al. (2015)	Quais são os desafios e limitações das técnicas de engenharia de requisitos em projeto ágeis?	<p>Requisitos não acordados com o time de desenvolvimento</p> <p>Falta de eficiência na análise e inspeção dos requisitos</p> <p>Baixa disponibilidade de ferramentas específicas para engenharia de requisitos ágil</p> <p>Falta de possibilidade para reusar requisitos e especificações</p> <p><i>Overscopping</i></p> <p>Necessário intensa comunicação com o cliente</p> <p>Dificuldade para incluir limitações técnicas nos cenários</p> <p>Esforço necessário para aprender a escrever bons cenários</p> <p>Cartões de histórias e o quadro são notações incompletas de um sistema</p> <p>Necessidade da presença do cliente para entender o quadro e os cartões de histórias</p> <p>TDD requer um profundo entendimento dos requisitos</p> <p>TDD requer colaboração extensiva entre o desenvolvedor e o cliente</p> <p>TDD envolve refinar especificações iterativamente em níveis mais baixos</p> <p>Casos de usuário possuem muita informação apresentada</p> <p>Diagramas (XXM, UC, Diag Ativ) são usados mais pelo time de design do que pelo cliente</p> <p>Nenhum dos diagramas foram identificados pelo cliente como úteis em encontrar falhas</p>

Tabela 9 – Problemas Comuns na Engenharia de Requisitos em Contextos Ágeis

Artigo	Pergunta de Pesquisa	Problemas citados
Medeiros et al. (2015)	Quais são os desafios e limitações das técnicas de engenharia de requisitos em projeto ágeis?	<p>Os diagramas (XXM, UC, Diag Ativ) eram difíceis de interpretar</p> <p>Estórias eram pequenas, vagas e ambíguas</p> <p>Habilidades técnicas são necessárias para escrever usável estórias de usuário</p> <p>Estórias de usuário - nível de detalhe não é apropriado, esforço considerado é obrigatório</p> <p>Estórias de usuário - inadequadas para descrever aspectos técnicos</p> <p>Tediosas sessões de escopo com os clientes</p> <p>Dificuldade em gerenciar <i>backlogs</i> grandes</p> <p>Manter SRS atualizado</p> <p>Esforço extra para integrar os requisitos</p> <p>Especificação de requisitos detalhada é produzida antecipadamente</p> <p>Falta de motivação do time por causa das mudanças constantes</p> <p>O controle na mudança de requisitos é insuficiente</p> <p>Arquiteturas não são escaláveis por causa das constantes mudanças</p> <p>Tempo gasto com mudanças de requisitos</p> <p>É difícil criar estimativas de custo, tempo e performance acuradas</p> <p>Re-priorização frequente dos requisitos</p> <p>Requisitos de várias origens geram muitos conflitos</p> <p>Falhas de comunicação</p> <p>Dificuldade com times distribuídos</p>

Tabela 9 – Problemas Comuns na Engenharia de Requisitos em Contextos Ágeis

Artigo	Pergunta de Pesquisa	Problemas citados
Medeiros et al. (2015)	Quais são os desafios e limitações das técnicas de engenharia de requisitos em projeto ágeis?	<p>Dificuldade em promover a sustentabilidade para os times</p> <p>Existe um número de falta de entendimento por causa da falta de pessoas chaves</p> <p>Relação não transparente entre os espaços do problema e da solução</p> <p>Requisitos não confiáveis, instáveis, superficiais e ambíguos</p> <p>Definição de requisitos não funcionais fraca</p> <p>Requisitos essenciais não são propriamente tratados</p>
Elghariani e Kama (2016)	Quais são os desafios da engenharia de requisitos com metodologias ágeis?	<p>Falta de documentação</p> <p>Disponibilidade do cliente</p> <p>Arquitetura de software inadequada</p> <p>Estimativas de orçamento e tempo para o projeto</p> <p>Ignorar requisitos não funcionais</p> <p>Mudança e reavaliação de requisitos</p>
Fernández et al. (2016)	Quais são os padrões de problemas observados e o contexto das características?	<p>Requisitos incompletos e/ou escondidos</p> <p>Falhas de comunicação entre o time do projeto e o cliente</p> <p>Mudança de alvos (mudança de objetivos, processos de negócio e/ou requisitos)</p> <p>Falha na especificação de requisitos que estão muito abstratos</p> <p><i>Time boxing</i>/em geral, falta de tempo</p> <p>Falhas de comunicação dentro do próprio projeto</p>

Tabela 9 – Problemas Comuns na Engenharia de Requisitos em Contextos Ágeis

Artigo	Pergunta de Pesquisa	Problemas citados
Fernández et al. (2016)	Quais são os padrões de problemas observados e o contexto das características?	<p><i>Stakeholders</i> com dificuldades em separar requisitos de soluções de design já conhecidas</p> <p>Requisitos inconsistentes</p> <p>Fraco acesso as necessidades do usuário</p>
Curcio et al. (2018)	Quais são os obstáculos que a engenharia de requisitos ágil enfrenta (ambiente, pessoas, recursos)?	<p>Dificuldades com comunicação em times distribuídos</p> <p>Dificuldades para encontrar específicas e especializadas habilidades</p> <p>Problemas de motivação relacionados ao trabalho de engenharia de requisitos</p> <p>Disponibilidade do cliente</p> <p>Falta de habilidade e acordo com o cliente</p> <p>Falta de precisão nas estimativas</p> <p>Falta de uma ferramenta especializada</p>

## APÊNDICE B – METODOLOGIA USADA NA REVISÃO SISTEMÁTICA DA LITERATURA

O presente documento apresenta a metodologia usada na condução da Revisão Sistemática da Literatura (RSL) cujo objetivo é identificar soluções para melhorar o entendimento dos requisitos pelo time. O resultado dessa revisão deu origem a lista de ações propostas que foram discutidas na seção 5.4.

Esse estudo foi conduzido como uma revisão sistemática da literatura baseado nas instruções propostas no *guideline* de Kitchenham e Charters (2007). Esse tipo de trabalho apresenta achados oriundos de outros artigos científicos. Por isso, ele é classificado como um estudo secundário.

### B.0.1 Pergunta de Pesquisa

A pergunta de pesquisa **RQ1** foi usada para guiar o processo de busca e a seleção dos estudos:

*RQ1: Quais são as ferramentas, técnicas ou abordagens que podem ser implementadas em projetos de desenvolvimento de software para tornar os requisitos mais completos para o time?*

Os achados encontrados na literatura e as discussões relacionadas a essa pergunta são apresentadas na subseção 6.1.1.

### B.0.2 Fonte dos Dados e Estratégia de Pesquisa

Como forma de contornar possíveis tendências entre grupo de artigos, uma busca automática foi executada no Google Scholar<sup>1</sup> usando a *string* de busca definida na Figura 19.

(incomplete OR immature OR unready OR high-level OR hidden)  
AND requirement AND “software development” AND agile AND  
(solution OR tool OR technique OR process OR approach)

Figura 19 – *String* de busca

A *string* de busca foi elaborada em inglês baseada nas palavras chaves identificadas nessa pesquisa: (1) incompleto, (2) requisitos, (3) desenvolvimento de software, (4) ágil e (5) solução. Em seguida, os sinônimos de alguns termos foram adicionados para expandir a abrangência da busca, conforme apresentado na Tabela 10. Durante pesquisas preliminares no engenho de busca notou-se que os termos (2), (3) e (4) eram comumente usados em todos os artigos. Por isso, não foram utilizados sinônimos para os mesmos.

<sup>1</sup><scholar.google.com.br>

Tabela 10 – Palavras-chaves e Sinônimos

Palavra-chave	Sinônimos
Incompleto	imaturo, não pronto, alto-nível e escondido
Solução	ferramenta, técnica, processo e abordagem

É importante mencionar que essa busca foi realizada em abril/2018 usando o modo de navegação anônimo do Google Chrome<sup>2</sup> para evitar que configurações pré-definidas interferissem no resultado da pesquisa. Por fim, foram coletados os cem artigos mais relevantes resultantes dessa pesquisa.

### B.0.3 Critérios de Inclusão e Exclusão

Foram incluídos todos os artigos que mencionavam o termo requisito no contexto de desenvolvimento de software. Em seguida, foram excluídos todos os artigos que atendem a qualquer um dos critérios de exclusão abaixo:

1. Estudos não escritos em inglês;
2. Não acessível na web;
3. Documentos incompletos, livros, rascunhos, slides de apresentações, resumos estendidos e estudos não primários (secundários e terciários);
4. Artigos que não apresentam nenhuma solução para tornar os requisitos de uma equipe mais completos;

### B.0.4 Seleção de Estudos

Os critérios de inclusão e exclusão definidos na seção acima (B.0.3) foram aplicados aos 100 artigos resultantes da busca no Google Scholar. Os artigos foram avaliados da seguinte forma: em primeiro lugar, os títulos, resumos e resultados eram lidos para identificar os artigos que não eram relevantes para a pesquisa. Em outras palavras, o artigo não apresentava nenhuma solução para tornar os requisitos mais completos, conforme declarado na pergunta de pesquisa da Sub-seção B.0.1. Se não fosse possível obter uma decisão sobre o artigo nessa primeira leitura, então, todo o artigo era lido usando a técnica de *skimming*.

Ao final do processo apresentado acima, 23 artigos foram selecionados. Limitações e indicações sobre trabalhos futuros sobre essa etapa da pesquisa já foram apresentados nas seções 7.1 e 7.2.

<sup>2</sup><[www.google.com/chrome](http://www.google.com/chrome)>

### B.0.5 Extração de Dados

Os quatro itens apresentados abaixo representam um formulário que foi criado para guiar a extração dos artigos. Basicamente, os itens buscados durante a extração eram:

- **Objetivo** geral do artigo analisado;
- **Método de Pesquisa** aplicado no estudo;
- **Contexto da Pesquisa** ou características gerais sobre a pesquisa; por exemplo, número de respondentes (no caso de *surveys*);
- **Soluções** encontrados na literatura para tornar os requisitos mais completos.

Os achados eram destacados durante uma primeira leitura, mas não eram extraídos. Depois de passar por todos os artigos selecionados, uma nova leitura foi realizada visando confirmar, descartar ou encontrar novos itens nesses artigos. Caso o trecho destacado fosse confirmado como uma solução, o mesmo era extraído e adicionado a um documento do Google Docs <sup>3</sup>, conforme ilustrado na Figura 20.

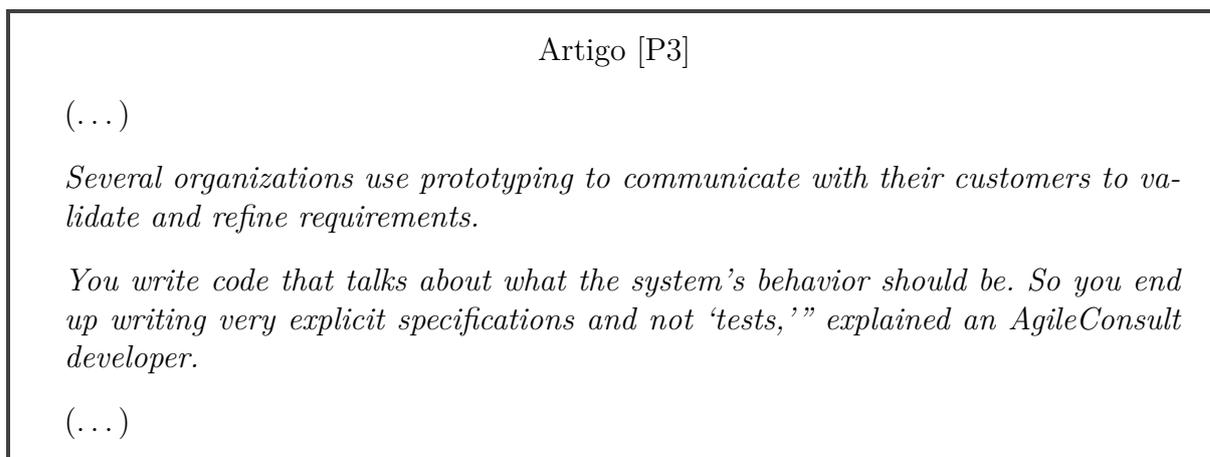


Figura 20 – Exemplo da Extração

Durante a extração verificou-se que os artigos [P3] e [P19] apresentam achados provenientes da mesma amostra de dados, um *survey* realizado em 16 empresas dos Estados Unidos. Sendo assim, os mesmos foram contados como apenas 1 estudo. É válido salientar que embora os dois artigos tenham o mesmo título e apresentem achados semelhantes, algumas possíveis ações estão presentes apenas no [P3] e outras apenas [P19]. Por isso, optou-se por manter os dois artigos.

Finalizada a extração, o documento com os dados em sua forma original estava com 21 páginas e um pouco mais de 5000 palavras.

<sup>3</sup><<https://docs.google.com/document>>

### B.0.6 Síntese dos Dados

Durante a etapa de síntese dos dados, o documento apresentado na Figura 20 foi importado no *software* MAXQDA<sup>4</sup> com o intuito de facilitar a análise dos dados. Optou-se por usar o *MAXQDA* porque ele é uma ferramenta voltada para análise qualitativa e facilita o processo de codificação aberta, que foi o tipo de codificação usada nessa revisão da literatura.

Por fim, depois de codificados, os itens foram agrupados em categorias para simplificar e otimizar a construção de *links* entre os itens relacionados (SEAMAN, 1999).

### ESTUDOS SELECIONADOS

- [P1] AMBLER, S. W. Agile software development at scale. In: *Balancing Agility and Formalism in Software Engineering*. Springer Berlin Heidelberg, 2008. p. 1–12. Disponível em: <[https://doi.org/10.1007/978-3-540-85279-7\\_1](https://doi.org/10.1007/978-3-540-85279-7_1)>.
- [P2] BJARNASON, E.; WNUK, K.; REGNELL, B. A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In: *Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11*. ACM Press, 2011. Disponível em: <<https://doi.org/10.1145/2068783.2068786>>.
- [P3] CAO, L.; RAMESH, B. Agile requirements engineering practices: An empirical study. *IEEE Software*, Institute of Electrical and Electronics Engineers (IEEE), v. 25, n. 1, p. 60–67, jan 2008. Disponível em: <<https://doi.org/10.1109/ms.2008.1>>.
- [P4] DAHLSTEDT, G.; KARLSSON, L.; NATT, J.; PERSSON, A. Requirements engineering challenges in market-driven software development – An interview study with practitioners. v. 49, p. 588–604, 2007.
- [P5] FOX, D.; SILLITO, J.; MAURER, F. Agile Methods and User-Centered Design : How These Two Methodologies Are Being Successfully Integrated In Industry. *Agile 2008 Conference*, IEEE, p. 63–72, 2008.
- [P6] HANSEN, G. K.; FÆGRI, T. E. Process fusion: An industrial case study on agile software product line engineering. v. 81, p. 843–854, 2008.
- [P7] HANSSON, C.; DITTRICH, Y.; ZARNAK, S. How agile are industrial software development practices ? v. 79, p. 1295–1311, 2006.
- [P8] HUO, M.; VERNER, J.; ZHU, L.; BABAR, M. A. Software Quality and Agile Methods. *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, IEEE, p. 520–525 vol.1, 2004.

---

<sup>4</sup><[www.maxqda.com](http://www.maxqda.com)>

- 
- [P9] KANE, D. W.; HOHMAN, M. M.; CERAMI, E. G.; MCCORMICK, M. W.; KUHLMAN, K. F.; BYRD, J. A. Agile methods in biomedical software development : a multi-site experience report. v. 12, p. 1–12, 2006.
- [P10] LEAU, Y.; LOO, W. K.; THAM, W. Y.; TAN, S. F. Software Development Life Cycle AGILE vs Traditional Approaches. v. 37, n. Icint, p. 162–167, 2012.
- [P11] LIN, J. Using Goal Net to Model User Stories in Agile Software Development. *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, p. 1–6, 2014.
- [P12] LUCIA, A. D.; QUSEF, A. Requirements Engineering in Agile Software Development. v. 2, n. 3, p. 212–220, 2010.
- [P13] MAURER, F.; MARTEL, S. Extreme programming. rapid development for web-based applications. *IEEE Internet Computing*, v. 6, n. 1, p. 86–90, Jan 2002. ISSN 1089-7801.
- [P14] MISHRA, D.; MISHRA, A. Complex software project development : agile methods adoption. n. January, p. 549–564, 2011.
- [P15] NORD, R. L.; TOMAYKO, J. E. Software architecture-centric methods and agile development. *IEEE Software*, v. 23, n. 2, p. 47–53, March 2006. ISSN 0740-7459.
- [P16] PAETSCH, F.; EBERLEIN, A.; MAURER, F. Requirements engineering and agile software development. In: *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003*. [S.l.: s.n.], 2003. p. 308–313. ISSN 1080-1383.
- [P17] PETERSEN, K.; WOHLIN, C. The Journal of Systems and Software A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *The Journal of Systems & Software*, Elsevier Inc., v. 82, n. 9, p. 1479–1490, 2009. ISSN 0164-1212. Disponível em: <<http://dx.doi.org/10.1016/j.jss.2009.03.036>>.
- [P18] PETERSEN, K.; WOHLIN, C. The effect of moving from a plan-driven to an incremental software development approach with agile practices An industrial case study. p. 654–693, 2010.
- [P19] RAMESH, B.; CAO, L.; BASKERVILLE, R. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, Wiley, v. 20, n. 5, p. 449–480, sep 2010.
- [P20] SAVOLAINEN, J.; KUUSELA, J.; VILAVAARA, A. Transition to agile development - rediscovery of important requirements engineering practices. In: *2010 18th IEEE Interna-*

---

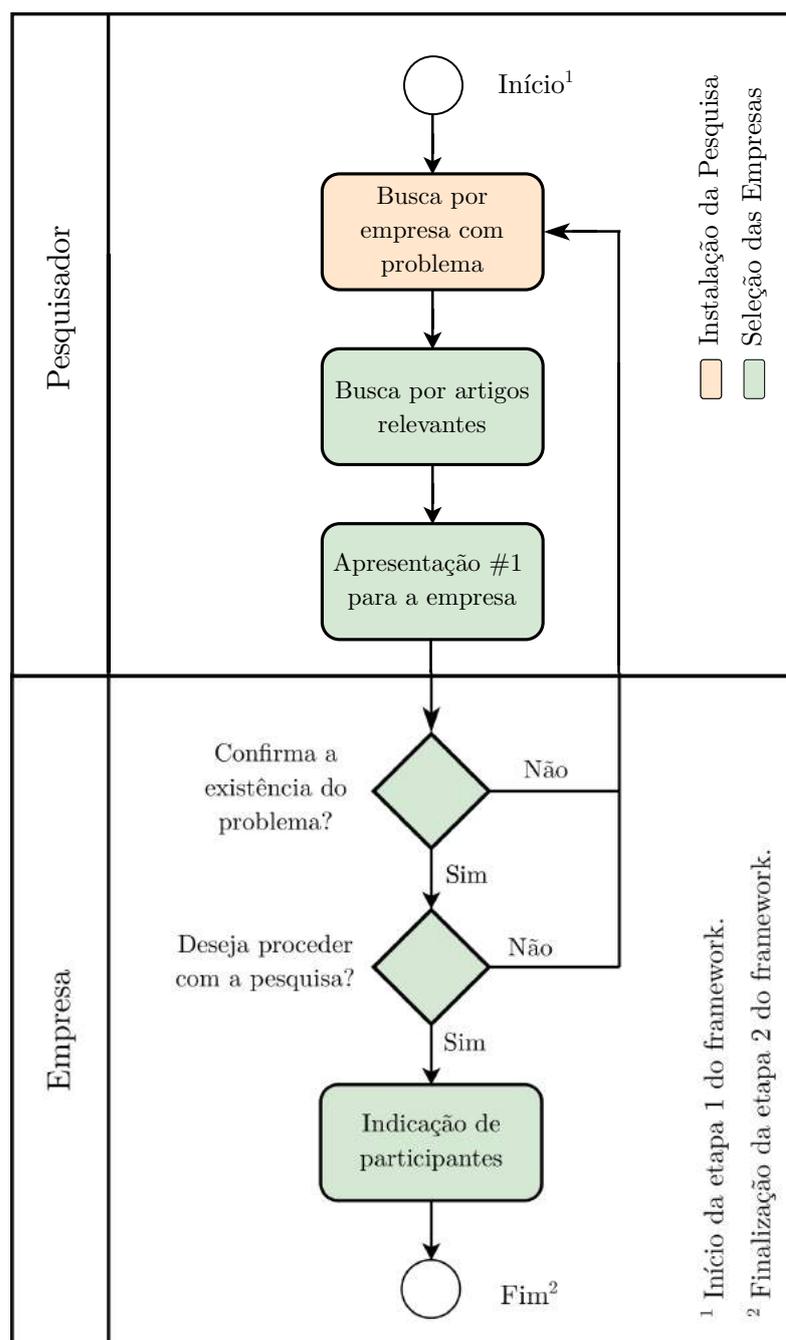
*tional Requirements Engineering Conference*. Sydney, Australia: [s.n.], 2010. p. 289–294. ISSN 2332-6441.

- [P21] STRODE, D. E.; HUFF, S. L.; HOPE, B.; LINK, S. The Journal of Systems and Software Coordination in co-located agile software development projects. *The Journal of Systems & Software*, Elsevier Inc., v. 85, n. 6, p. 1222–1238, 2012. ISSN 0164-1212. Disponível em: <<http://dx.doi.org/10.1016/j.jss.2012.02.017>>.
- [P22] TURK, D.; FRANCE, R. B.; RUMPE, B. Assumptions underlying agile software development processes. *CoRR*, abs/1409.6610, 2014. Disponível em: <<http://arxiv.org/abs/1409.6610>>.
- [P23] VLAANDEREN, K.; JANSEN, S.; BRINKKEMPER, S.; JASPERS, E. The agile requirements refinery : Applying SCRUM principles to software product management. *Information and Software Technology*, Elsevier B.V., v. 53, n. 1, p. 58–70, 2011. ISSN 0950-5849. Disponível em: <<http://dx.doi.org/10.1016/j.infsof.2010.08.004>>.

## APÊNDICE C – FRAMEWORK PARA PESQUISA-AÇÃO NA INDÚSTRIA

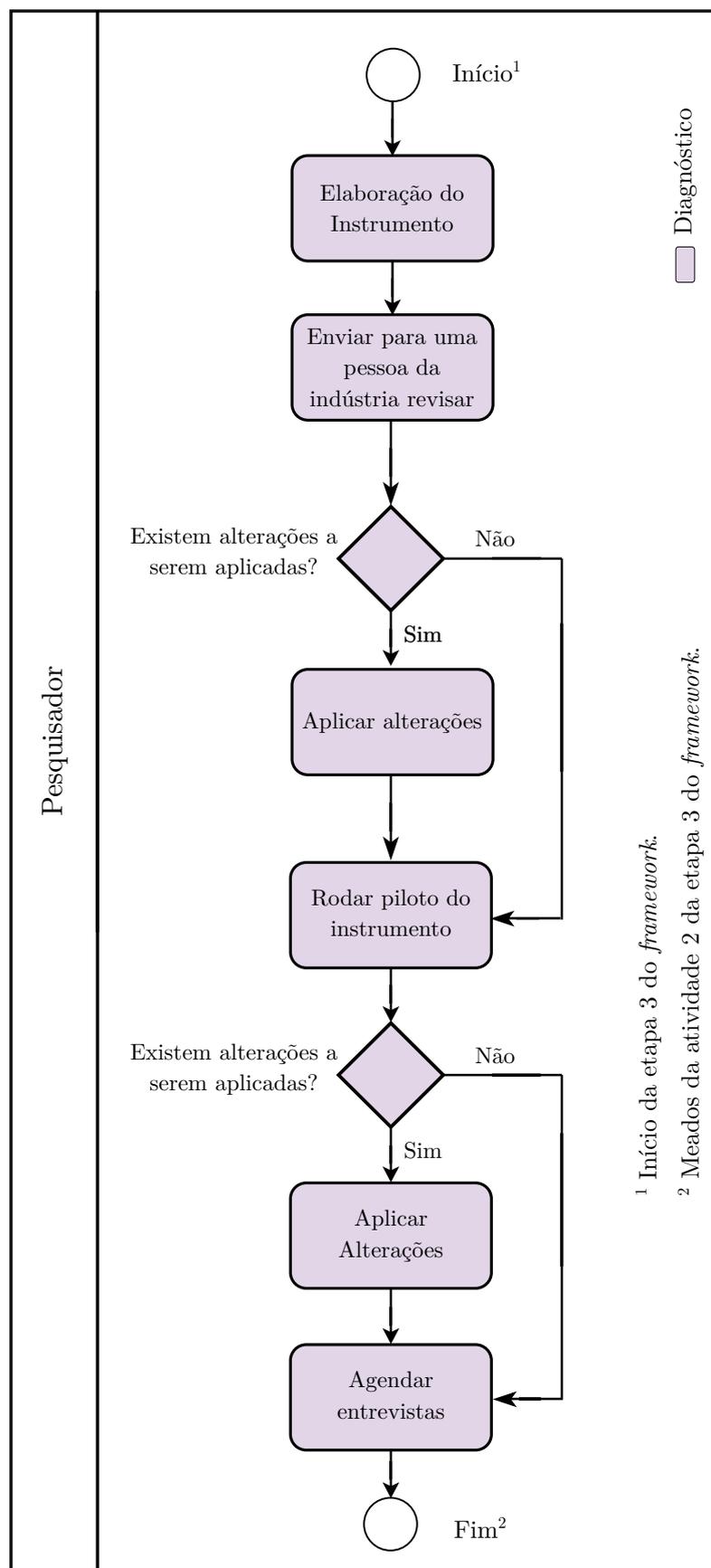
Esse documento apresenta um conjunto de fluxos que servem para guiar a condução de uma pesquisa-ação na indústria. Os mesmos foram desenvolvidos durante a construção do método CReq-SW e baseados no trabalho de Suassuna (2017).

Figura 21 – Fluxograma das Etapas de Seleção das Empresas e Instalação da Pesquisa



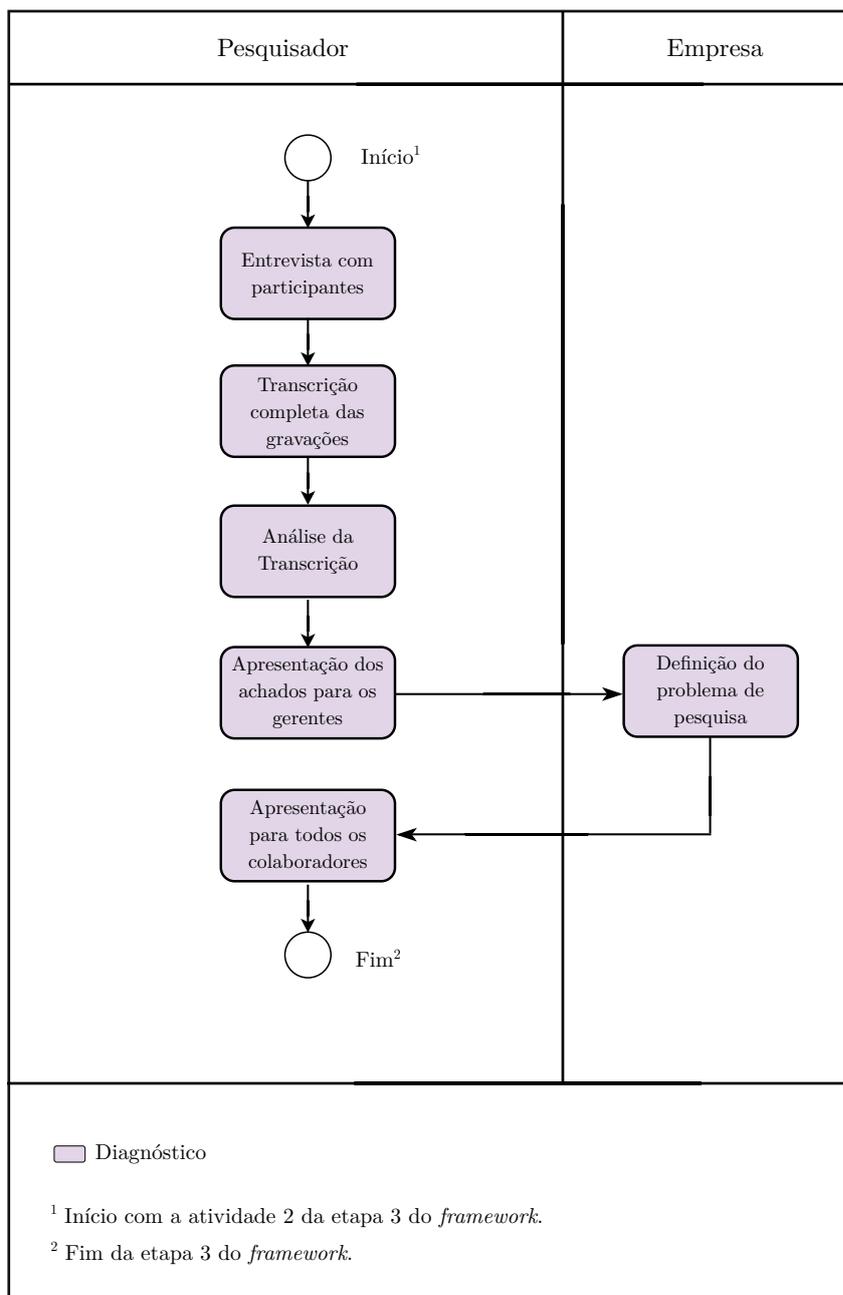
Fonte: Elaboração Própria

Figura 22 – Fluxograma das Atividades 1 e 2 (parcial) da Etapa de Diagnóstico



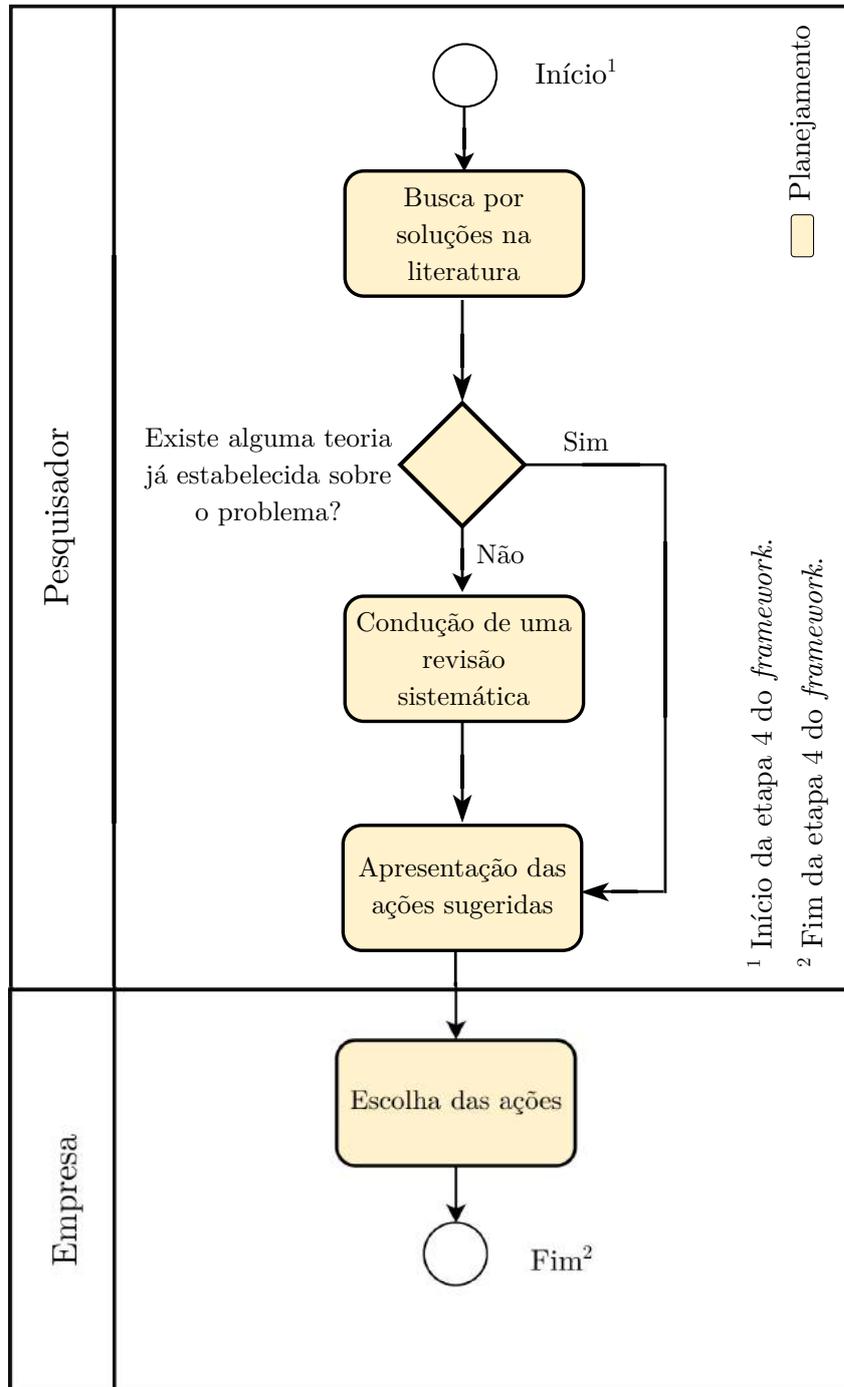
Fonte: Elaboração Própria

Figura 23 – Fluxograma das Atividades 2, 3, 4, 5 e 6 da Etapa de Diagnóstico



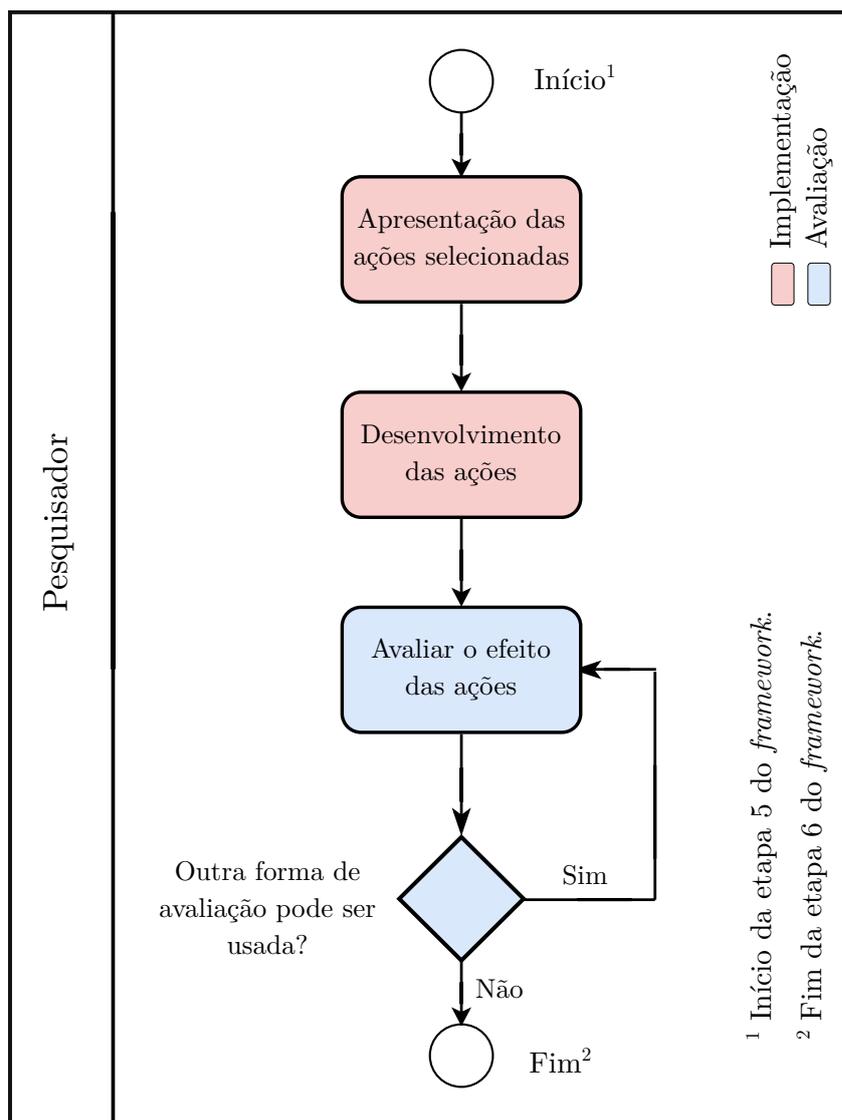
Fonte: Elaboração Própria

Figura 24 – Fluxograma da Etapa de Planejamento



Fonte: Elaboração Própria

Figura 25 – Fluxograma das Etapas de Implementação e Avaliação



Fonte: Elaboração Própria

## APÊNDICE D – ROTEIRO DAS ENTREVISTAS

Roteiro utilizado nas entrevistas da etapa de diagnóstico da pesquisa-ação, conforme descrito na seção 5.3.

### D.1 *CHECKLIST* INÍCIO

- Apresentação pesquisador;
- Objetivo da pesquisa;
- Introdução sobre pesquisa-ação;
- Propósito da entrevista;
- Observações gerais:
  - Máximo de detalhes possível;
  - Duração média de 30 minutos;
  - Permissão para gravar;
  - A entrevista e gravação podem ser interrompidas a qualquer momento, caso seja desejado;

### D.2 IDENTIFICAÇÃO DO ENTREVISTADO

- Nome
- Papel / função
- Tempo de experiência
- Tempo na empresa

### D.3 INTRODUÇÃO AO PROJETO

- Qual o nome do projeto?
- Como a equipe é formada?
  - Quantas pessoas?
  - Qual o horário normal de trabalho?
  - É uma equipe multidisciplinar?

- Como a equipe é dividida? Existe o conceito de sub-times?
- Segue alguma metodologia ou *framework* ágil? Se sim, qual?
  - \* Quais práticas ágeis a equipe já adotou?

#### D.4 CLIENTE

- Quem é o cliente do projeto?
  - Ele está on-site?
  - Faz parte do time? Está envolvido no processo?
- Como é feita a comunicação com ele? Quais meios são usados?
- Toda equipe tem liberdade para conversar/questioná-lo?

#### D.5 PROCESSO DE DESENVOLVIMENTO

- Como é o ciclo de desenvolvimento de vocês?
  - As iterações são de quantos dias?
- O escopo é bem definido?
- O time costuma receber demandas ou já recebeu demandas não planejadas? Com que frequência? De onde essas mudanças vem? Como o time lida com isso?
- Como é feita a priorização? Quem cuida do *backlog*?
- E as cerimônias? Quais são conduzidas? Quem participa?
  - Como o *planning* é feito?
  - Como o cliente valida o que foi feito? Quando?

##### D.5.1 Requisitos

- Como os requisitos chegam até você?
- Onde eles ficam documentados?
  - O time costuma usar esse documento e receber atualizações quando eles são modificados?
  - Você consegue entendê-los com facilidade?
  - Como as mudanças são trackeadas?
- Quais técnicas são usadas na elicitação de requisitos?
- Requisitos não funcionais são levados em consideração?

### D.5.2 Testes

- Como é feita a validação e verificação dos requisitos?
- Você sente dificuldade em manter as suítes de testes sempre atualizadas?
- É comum ter defeitos escapados? É comum achar defeitos nos caminho feliz?

### D.6 *CHECKLIST* CONCLUSÃO

- Esqueci de perguntar algo relevante sobre o seu projeto relacionado a esse tema?
- Agradecer a participação;
- Disponibilizar e-mail para contato;

## APÊNDICE E – SLIDES DA APRESENTAÇÃO 1 - PROBLEMAS COMUNS



Balasubramaniam Ramesh, Lan Cao e Richard Baskerville  
DOI: [10.1111/j.1365-2575.2007.00259.x](https://doi.org/10.1111/j.1365-2575.2007.00259.x)  
**Agile Requirements Engineering Practices and Challenges: An Empirical Study**

Cin.utpe.br



### Agenda

- Motivação
- Método
- Resultados
  - Práticas
  - Desafios
- Conclusão

Cin.utpe.br



### Motivação

*“Most software development organizations are forced to deal with requirements that tend to evolve very quickly and become obsolete even before project completion.”*

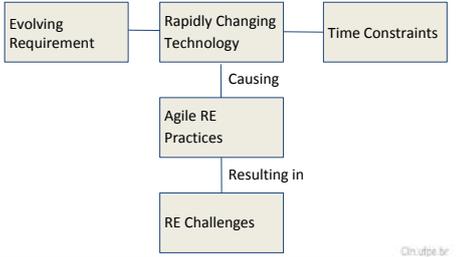
(Boehm, 2000)

Agile methods and its practices.  
(Agile Alliance, 2001)

Cin.utpe.br



### Framework de Pesquisa



```

graph TD
    A[Evolving Requirement] --- B[Rapidly Changing Technology]
    B --- C[Time Constraints]
    B -- Causing --> D[Agile RE Practices]
    D -- Resulting in --> E[RE Challenges]
  
```

Cin.utpe.br



### Método

- Múltiplos casos (16 organizações);
- Duas fases (envolvidos com ágil e adeptos de um método específico);
- Uso de entrevistas semi-estruturadas, observação dos participantes e revisão de documentação;
- **Objetivo:** entender como e porque a análise de requisitos em contextos ágeis difere do tradicional

Cin.utpe.br



### Resultados - Práticas

- **Comunicação face a face mais do que especificação escrita:** o foco da engenharia de requisitos em um contexto ágil é para transferir ideias do cliente para o time de desenvolvimento, ao invés de criar extensivos documentos de requisitos.
- **Engenharia de requisitos iterativa:** a análise de requisitos ágeis deve continuar em cada ciclo de desenvolvimento.
- **Priorização de requisitos ao extremo:** para manter as prioridades atualizadas, a priorização é repetida frequentemente durante todo o processo de desenvolvimento.

Cin.utpe.br



## Resultados - Práticas

- **Gerenciar mudanças de requisitos através do planejamento constante:** um princípio *core* no desenvolvimento de software ágil é a adaptação e a rápida resposta às mudanças originadas pelo ambiente.
- **Uso de protótipos:** invés de usar documentos formais de requisitos, muitos projetos usam protótipos como uma forma de se comunicar com os clientes.
- **Uso de reuniões de *review* e testes de aceitação:** as reuniões de *review* mostram que o projeto está na direção certa dentro do tempo estimado. Isso aumenta a confiança do cliente no time. Já os testes de aceitação são desenvolvidos pelo cliente, algumas vezes com a ajuda de QA, são outro meio usado para validação e verificação.

[cin.ufpe.br](http://cin.ufpe.br)



## Resultados - Desafios

- **Problemas com estimativas de custo e tempo:** é difícil desenvolver estimativas acuradas de tempo durante os estágios iniciais de um projeto em um contexto ágil.
- **Inadequada ou inapropriada arquitetura:** a arquitetura escolhida pelo time de desenvolvimento durante os primeiros ciclos pode tornar-se inadequada ou inapropriada a medida que novos requisitos vão sendo conhecidos.
- **Negligenciamento de requisitos não funcionais:** falta de atenção aos requisitos não funcionais. Geralmente, eles são mal definidos e ignorados durante os ciclos iniciais de desenvolvimento.

[cin.ufpe.br](http://cin.ufpe.br)



## Resultados - Desafios

- **Acesso ao cliente e participação:** a efetividade da comunicação entre o cliente e o time depende de alguns fatores. Entre eles: disponibilidade do cliente, confiança, consenso, etc.
- **Priorização em uma dimensão apenas:** uso de valor de negócio como critério único ou primário para priorização de requisitos pode trazer problemas.
- **Verificação inadequada dos requisitos:** algumas vezes, a falta de clareza ou a falta de especificações completas de requisitos críticos, pode tornar mais difícil o processo de desenvolvimento.
- **Documentação mínima:** a falta de documentação pode gerar uma variedade de problemas relacionados a falhas de comunicação.

[cin.ufpe.br](http://cin.ufpe.br)



Obrigada!  
Alguma **dúvida?**

Karla Silva - [kmbs@cin.ufpe.br](mailto:kmbs@cin.ufpe.br)

[cin.ufpe.br](http://cin.ufpe.br)

## APÊNDICE F – SLIDES DA APRESENTAÇÃO 2 - DIAGNÓSTICO

### F.0.1 Versão Gerentes

 <p>Projeto de <b>Pesquisa</b> CIn/Rec Tecnologia &amp; UFPE</p> <p>Programa de Pós-Graduação Karla Silva - kmbs@cin.ufpe.br</p> <p><small>Cin.ufpe.br</small></p>	 <h4>Roteiro das Entrevistas</h4> <ul style="list-style-type: none"> <li>• Introdução ao projeto;</li> <li>• Cliente;</li> <li>• Processo de desenvolvimento;             <ul style="list-style-type: none"> <li>- Requisitos;</li> <li>- Testes;</li> </ul> </li> </ul> <p><small>Cin.ufpe.br</small></p>
---	---

 <h4>Agrupamento</h4> <p>Os achados foram agrupados em dois grupos:</p> <ul style="list-style-type: none"> <li>• <b>Oportunidades de melhoria:</b> Práticas que são consideradas “dores” para o time.</li> <li>• <b>Pontos fortes:</b> Práticas que estão funcionando muito bem no time e que serviria de exemplo para outros.</li> </ul> <p><small>Cin.ufpe.br</small></p>	 <h4>Projeto A13</h4> <p>Entrevistas realizadas no período de 25/out a 16/nov; Quatro pessoas entrevistadas (gerente, desenvolvedor, tester e designer); Aprox. 123 minutos de entrevistas*;</p> <p><small>Cin.ufpe.br</small></p>
--	---

 <h4>Oportunidades de Melhoria</h4> <ul style="list-style-type: none"> <li>• Os requisitos chegam sempre em bem alto-nível ou são muito complexos;</li> <li>• Os designers do projeto na UFPE <b>só consultam</b> os requisitos disponibilizados pelo cliente, <b>não o alteram</b>;</li> <li>• O time costuma receber solicitações de mudanças durante a sprint;</li> <li>• <b>Normalmente</b>, quem tem mais contato com o cliente é o Project Manager;</li> </ul> <p><small>Cin.ufpe.br</small></p>	 <h4>Oportunidades de Melhoria</h4> <ul style="list-style-type: none"> <li>• Quando as dúvidas são sanadas, <b>algumas vezes</b>, os requisitos são atualizados;</li> <li>• No entanto, <b>não existe tracking</b> entre as stories (dependências);</li> <li>• A documentação de UX deveria ficar pronta antes da sprint de dev, mas devido às mudanças constantes, nem sempre isso acontece;</li> </ul> <p><small>Cin.ufpe.br</small></p>
---	---



### Oportunidades de Melhoria

- UX trabalha na **projeção da solução sozinho**. Às vezes, os devs encontram problemas lá na frente já;
- Às vezes, o time não tem visão nenhuma sobre o futuro do backlog;
- O time em si **não se preocupa muito** com requisitos não funcionais, mas seguem o que foi proposto pelo designer com relação a usabilidade;

Cin.utpe.br



### Oportunidades de Melhoria

- O time tem abertura para estimar e fazer sugestões de priorização, **mas às vezes**, a decisão já foi tomada pelo cliente;
- Negociações sobre o escopo da sprint **não são tão tranquilas**, mas isso é mais um trabalho PM. O time **já recebe** o que tem que ser feito;
- Ser um projeto piloto em vários âmbitos (approaches diferentes de outros projetos);

Cin.utpe.br



### Pontos Fortes

- Durante o planning, tem uma rodada de dúvidas para refinar a estória, e o cliente responde-às em algum momento;
- O time utiliza a prática de code-review com pelo menos 2 pessoas;
- O time não sente dificuldades em definir aspectos relacionados à arquitetura; pois, a arquitetura foi definida pelos mesmos;

Cin.utpe.br



### Pontos Fortes

- A cada sprint, uma página de “**Decisions Log**” no Confluence é criada para que decisões importantes da sprint sejam catalogadas;
- Em momentos de incertezas sobre approaches, o time já usou algumas sprints como prova de conceito;

Cin.utpe.br



### Projeto B12

Entrevistas realizadas no período de 25/out a 16/nov;  
Três pessoas entrevistadas (gerente, desenvolvedor e tester);  
Aprox. 68 minutos de entrevistas;

Cin.utpe.br



### Oportunidades de Melhoria

- Para a Rec Tecnologia, os requisitos chegam um **pouco mais detalhado** (GTI-Br já tem feito uma refinada), mas ainda é considerado muito aberto;
- O cliente **sempre muda de opinião**, mas a equipe do GTI-Br tenta **ajudar** a ajustar o escopo;
- Geralmente, o time só consegue ler as stories quando elas estão para ser implementadas;

Cin.utpe.br



### Oportunidades de Melhoria

- Às vezes ocorrem atrasos no desenvolvimento, e isso compromete alguns dias de teste;
- Durante a sprint são testadas **apenas as stories da sprint**; Ao final, é que o todo é testado;
- Não existe comunicação direta com a *GlobeTech* (gerente do GTI-Br que passa as demandas).

Cin.ufpe.br



### Pontos Fortes

- Comunicação **intensa e contínua** com o time do GTI-Br;
- O cliente é visto como um parceiro (no sucesso ou na falha);
- Os devs podem visualizar os casos de testes durante o desenvolvimento; evitando assim, possíveis bugs.
- Existe um grande backlog já priorizado pelo time da *GlobeTech*;

Cin.ufpe.br



### Pontos Fortes

- Qualquer decisão de arquitetura, design e etc é feita em conjunto com a equipe da GTI-Br (extensão da equipe)
- Mudanças de requisitos são priorizadas de acordo com a necessidade atual (rebalancear escopo atual);
- Mudanças de requisitos são comunicadas ao time e atualizadas nos testes e regressão.
- É incomum haver mudanças de escopo durante a sprint;

Cin.ufpe.br



### Pontos Fortes

- Antes de ir para produção, o time roda *smoked* testes;
- O time tem visibilidade do backlog futuro, mas entende que ele é volátil e pode mudar;
- O time tem abertura para opinar na priorização dos requisitos \*;
- O time tem espaço e gosta de fazer refactoring;

Cin.ufpe.br



### Referências

- Challenges in requirements engineering - 10.1109/ISRE.1995.512557
- Requirements Researchers: Do We Practice What We Preach?
- Requirements engineering as a success factor in software projects - 10.1109/MS.2001.936219
- Naming the pain in requirements engineering - 10.1007/s10664-016-9451-7

Cin.ufpe.br



Obrigada!  
Alguma **dúvida**?

Karla Silva - kmbs@cin.ufpe.br

Cin.ufpe.br

## F.0.2 Versão Colaboradores



**Projeto de Pesquisa**  
CIn/Rec Tecnologia & UFPE

Programa de Pós-Graduação  
Karla Silva - kmbs@cin.ufpe.br

Cin.ufpe.br



Engenharia de Requisitos. Dois projetos usados na etapa de diagnóstico.  
Duração prevista entre 10 e 11 meses.

**Introdução**

Cin.ufpe.br



Pesquisa-ação. Artigos relacionados com o trabalho.

**Fundamentação Teórica**

Cin.ufpe.br



*A pesquisa ação caracteriza-se por propor e implantar mudanças em situações previamente estudadas com o objetivo de **melhorá-las**.*

Steve Easterbrook (2008)

Cin.ufpe.br



**Ciclos da Pesquisa-ação**

- **Diagnóstico:** fase relacionada à identificação do problema que gerou o desejo de mudança pela organização;
- **Planejamento:** pesquisadores e colaboradores baseados em teorias elaboram um plano com as ações que devem ser aplicadas na empresa dona do problema;
- **Implementação:** de forma colaborativa, pesquisadores e colaboradores trabalham na implementação das ações definidas na etapa anterior;

Cin.ufpe.br



**Ciclos da Pesquisa-ação**

- **Avaliação:** depois de todas as ações terem sido executadas, é necessário avaliar se os efeitos teóricos propostos foram atingidos ou não;
- **Aprendizagem:** o processo de reflexão sobre o ciclo pode e deve ser executado permeando todas as outras fases. Não necessitando assim ser apenas a última fase;

Cin.ufpe.br



## Fundamentação Teórica

- Engenharia de requisitos;
- Dois projetos usados no diagnóstico;
- Duração estimada entre 10 e 11 meses;

cin.utpe.br



## Roteiro das Entrevistas

- Introdução ao projeto;
- Cliente;
- Processo de desenvolvimento;
  - Requisitos;
  - Testes;

cin.utpe.br



## Artigos Relacionados

Naming the pain in requirements engineering:  
Contemporary problems, causes, and effects in  
practice.

- D. Mendez Fernandez e outros
- DOI: 10.1007/s10664-016-9451-7
- 2017

cin.utpe.br



## Artigos Relacionados

Agile requirements engineering practices and  
challenges: an empirical study

- Balasubramaniam Ramesh e outros;
- DOI: 10.1111/j.1365-2575.2007.00259.x
- 2010

cin.utpe.br



## Agrupamento

Os achados foram agrupados em dois grupos:

- **Oportunidades de melhoria:** Práticas que são consideradas “dores” para o time.
- **Pontos fortes:** Práticas que estão funcionando muito bem no time e que serviria de exemplo para outros.

cin.utpe.br



## Projeto A13

Entrevistas realizadas no período de 25/out a 16/nov;  
Quatro pessoas entrevistadas (gerente, desenvolvedor,  
tester e designer);  
Aprox. 123 minutos de entrevistas\*;

cin.utpe.br



### Oportunidades de Melhoria

- Os requisitos chegam sempre em bem alto-nível ou são muito complexos;
- Os designers do projeto na UFPE **só consultam** os requisitos disponibilizados pelo cliente, **não o alteram**;
- O time costuma receber solicitações de mudanças durante a sprint;
- **Normalmente**, quem tem mais contato com o cliente é o Project Manager;

[cin.ufpe.br](http://cin.ufpe.br)



### Oportunidades de Melhoria

- Quando as dúvidas são sanadas, **algumas vezes**, os requisitos são atualizados;
- No entanto, **não existe tracking** entre as stories (dependências);
- A documentação de UX deveria ficar pronta antes da sprint de dev, mas devido às mudanças constantes, nem sempre isso acontece;

[cin.ufpe.br](http://cin.ufpe.br)



### Oportunidades de Melhoria

- UX trabalha na **projeção da solução sozinho**. Às vezes, os devs encontram problemas lá na frente já;
- Às vezes, o time não tem visão nenhuma sobre o futuro do backlog;
- O time em si **não se preocupa muito** com requisitos não funcionais, mas seguem o que foi proposto pelo designer com relação a usabilidade;

[cin.ufpe.br](http://cin.ufpe.br)



### Oportunidades de Melhoria

- O time tem abertura para estimar e fazer sugestões de priorização, **mas às vezes**, a decisão já foi tomada pelo cliente;
- Negociações sobre o escopo da sprint **não são tão tranquilas**, mas isso é mais um trabalho PM. O time **já recebe** o que tem que ser feito;
- Ser um projeto piloto em vários âmbitos (approaches diferentes de outros projetos);

[cin.ufpe.br](http://cin.ufpe.br)



### Pontos Fortes

- Durante o planning, tem uma rodada de dúvidas para refinar a estória, e o cliente responde-às em algum momento;
- O time utiliza a prática de code-review com pelo menos 2 pessoas;
- O time não sente dificuldades em definir aspectos relacionados à arquitetura; pois, a arquitetura foi definida pelos mesmos;

[cin.ufpe.br](http://cin.ufpe.br)



### Pontos Fortes

- A cada sprint, uma página de "**Decisions Log**" no Confluence é criada para que decisões importantes da sprint sejam catalogadas;
- Em momentos de incertezas sobre approaches, o time já usou algumas sprints como prova de conceito;

[cin.ufpe.br](http://cin.ufpe.br)



## Projeto B12

Entrevistas realizadas no período de 25/out a 16/nov;  
Três pessoas entrevistadas (gerente, desenvolvedor e tester);  
Aprox. 68 minutos de entrevistas;

[Cin.utpe.br](http://Cin.utpe.br)



## Oportunidades de Melhoria

- Para a Rec Tecnologia, os requisitos chegam um **pouco mais detalhado** (GTI-Br já tem feito uma refinada), mas ainda é considerado muito aberto;
- O cliente **sempre muda de opinião**, mas a equipe do GTI-Br tenta **ajudar** a ajustar o escopo;
- Geralmente, o time só consegue ler as stories quando elas estão para ser implementadas;

[Cin.utpe.br](http://Cin.utpe.br)



## Oportunidades de Melhoria

- Às vezes ocorrem atrasos no desenvolvimento, e isso compromete alguns dias de teste;
- Durante a sprint são testadas **apenas as stories da sprint**; Ao final, é que o todo é testado;
- Não existe comunicação direta com a *GlobeTech* (gerente do GTI-Br que passa as demandas).

[Cin.utpe.br](http://Cin.utpe.br)



## Pontos Fortes

- Comunicação **intensa** e **contínua** com o time do GTI-Br;
- O cliente é visto como um parceiro (no sucesso ou na falha);
- Os devs podem visualizar os casos de testes durante o desenvolvimento; evitando assim, possíveis bugs.
- Existe um grande backlog já priorizado pelo time da *GlobeTech*;

[Cin.utpe.br](http://Cin.utpe.br)



## Pontos Fortes

- Qualquer decisão de arquitetura, design e etc é feita em conjunto com a equipe da GTI-Br (extensão da equipe)
- Mudanças de requisitos são priorizadas de acordo com a necessidade atual (rebalancear escopo atual);
- Mudanças de requisitos são comunicadas ao time e atualizadas nos testes e regressão.
- É incomum haver mudanças de escopo durante a sprint;

[Cin.utpe.br](http://Cin.utpe.br)



## Pontos Fortes

- Antes de ir para produção, o time roda *smoked* testes;
- O time tem visibilidade do backlog futuro, mas entende que ele é volátil e pode mudar;
- O time tem abertura para opinar na priorização dos requisitos\*;
- O time tem espaço e gosta de fazer refactoring;

[Cin.utpe.br](http://Cin.utpe.br)



Rastreabilidade entre estórias

Requisitos Complexos

**Requisitos incompletos**

Visão do Backlog

Projeção da Solução feita só por UX

Atrasos de Entregas

Problemas de Comunicação

Mudanças de requisitos

Projeto Piloto

Falta de Autonomia

Sem Tempo para Testes

[Cin.ufpe.br](http://Cin.ufpe.br)



## Referências

- Challenges in requirements engineering - 10.1109/ISRE.1995.512557
- Requirements Researchers: Do We Practice What We Preach?
- Requirements engineering as a success factor in software projects - 10.1109/MS.2001.936219
- Naming the pain in requirements engineering - 10.1007/s10664-016-9451-7

[Cin.ufpe.br](http://Cin.ufpe.br)



Obrigada!  
Alguma **dúvida?**

Karla Silva - kmbs@cin.ufpe.br

[Cin.ufpe.br](http://Cin.ufpe.br)

## APÊNDICE G – SLIDES DA APRESENTAÇÃO 3 - APRESENTAÇÃO DAS POSSÍVEIS AÇÕES

 <p>Projeto de <b>Pesquisa</b> CIn/Rec Tecnologia &amp; UFPE</p> <p>Programa de Pós-Graduação Karla Silva - kmbs@cin.ufpe.br</p> <p><small>Cin.ufpe.br</small></p>	 <p>Método e problemas de pesquisa.</p> <p><b>Visão Geral</b></p> <p><small>Cin.ufpe.br</small></p>
 <p><i>A pesquisa ação caracteriza-se por propor e implantar mudanças em situações previamente estudadas com o objetivo de <b>melhorá-las</b>.</i></p> <p>Steve Easterbrook (2008)</p> <p><small>Cin.ufpe.br</small></p>	 <p>Rastreabilidade entre estórias      Requisitos Complexos</p> <p><b>Requisitos incompletos</b>      Visão do Backlog</p> <p>Projeção da Solução feita só por UX</p> <p>Problemas de Comunicação      Atrasos de Entregas</p> <p>Projeto Piloto      Mudanças de requisitos</p> <p>Sem Tempo para Testes      Falta de Autonomia</p> <p><small>Cin.ufpe.br</small></p>
 <p>Soluções encontradas na literatura.</p> <p><b>Literatura</b></p> <p><small>Cin.ufpe.br</small></p>	 <p><b>Achados da Literatura</b></p> <ul style="list-style-type: none"> <li>• <b>Forte envolvimento do cliente:</b> O cliente precisa está on-site ou disponível para o time e inserido em todo o processo de desenvolvimento do produto. Assim, ele pode oferecer <i>feedbacks</i> mais cedo.</li> <li>• <b>Papel de representante do cliente:</b> Se o cliente não tem disponibilidade, uma pessoa <b>empoderada</b> para tomar decisões pode assumir esse papel.</li> </ul> <p><small>Cin.ufpe.br</small></p>



### Achados da Literatura

- **Canal aberto com o cliente:** Uso de ferramentas de comunicação instantânea e vídeo conferências e comunicação face-a-face.
- **Daily com o cliente:** Reuniões diárias de acompanhamento de status com a presença do "cliente".

[On.utpe.br](http://On.utpe.br)



### Achados da Literatura

- **Sessões de *Backlog Refinement (grooming)*:** Antes de cada ciclo de desenvolvimento o cliente apresenta um conjunto de features para a próxima iteração.
- **Sprint Review:** O time tem a oportunidade de mostrar o que foi desenvolvido durante a sprint e obter o feedback do cliente.

[On.utpe.br](http://On.utpe.br)



### Achados da Literatura

- **Participação na escrita de Testes de Aceitação:** O cliente pode ajudar na escrita (ou validação) dos testes de aceitação.
- **Rastreabilidade:** O rastreamento entre casos de testes e requisitos pode ser útil como documentação e para entender como o código evoluiu ao longo do tempo.

[On.utpe.br](http://On.utpe.br)



### Achados da Literatura

- **TDD:** A adoção do TDD (*Test Driven Development*) ajuda a explorar cenários antes da codificação em si.
- **Escrita de testes para capturar requisitos:** Os cenários de teste devem ser levantados o mais cedo possível; por exemplo, durante o *planning*.

[On.utpe.br](http://On.utpe.br)



### Achados da Literatura

- **Proximidade dos membros do time:** Designers e Testers trabalham juntos com o time de desenvolvimento. Idealmente, eles devem ser próximos para acelerar o ciclo de feedback.
- **Priorização constante:** O backlog deve ser priorizado constantemente, principalmente antes do início de cada sprint.

[On.utpe.br](http://On.utpe.br)



### Achados da Literatura

- **Releases Pilotos:** Alguns usuários podem participar de releases pilotos para a validação de determinada feature ou testes com o usuários.
- **Releases Frequentes:** Releases frequentes ajudam a ter ciclos de *feedback* curtos.

[On.utpe.br](http://On.utpe.br)



## Achados da Literatura

- **Quebra de Requisitos:** O time pode negociar com o cliente quando achar que uma *user story* não possui o tamanho adequado. Além disso, escopo reduzido ou iterações menores também podem ser adotados.
- **Uso de Protótipos:** Protótipos ajudam o time e o cliente a ter uma visão sobre o que vai ser desenvolvido.

[cin.ufpe.br](http://cin.ufpe.br)



## Achados da Literatura

- **Uso de linguagem natural ou métodos linguísticos:** Requisitos devem ser coletados usando uma linguagem comum que qualquer pessoa naquele contexto consiga entender.
- **Uso de diferentes técnicas para elicitación e documentação:** técnicas como observação e entrevistas podem ser usadas.
- **Uso de User Stories:** Adoção de ferramentas para as histórias de usuário em formato eletrônico.

[cin.ufpe.br](http://cin.ufpe.br)



## Achados da Literatura

- **Uso de UML:** Uso da modelagem UML para descrever uma descrição alto nível da aplicação ou criar documentação do código.
- **Uso de modelos:** Por exemplo, *Initial Goal Net* para facilitar o entendimento dos requisitos pelo time.

[cin.ufpe.br](http://cin.ufpe.br)



Quais dessas práticas e/ou ideias podemos adotar/adaptar?

## Contexto Rec Tecnologia

[cin.ufpe.br](http://cin.ufpe.br)



Obrigada!  
Alguma dúvida?

Karla Silva - [kmbs@cin.ufpe.br](mailto:kmbs@cin.ufpe.br)

[cin.ufpe.br](http://cin.ufpe.br)

## APÊNDICE H – SLIDES DA APRESENTAÇÃO 4 - AÇÕES ESCOLHIDAS

 <p>Projeto de <b>Pesquisa</b> CIn/Rec Tecnologia &amp; UFPE</p> <p>Programa de Pós-Graduação Karla Silva - kmbs@cin.ufpe.br</p> <p style="text-align: right;"><small>Cin.ufpe.br</small></p>	 <h3>Ações Escolhidas</h3> <ol style="list-style-type: none"> <li>1. <i>Product Backlog Refinement</i> (grooming);</li> <li>2. <i>Sprint Review</i>;</li> <li>3. Participação na escrita de testes de aceitação;</li> <li>4. Rastreabilidade entre casos de testes e código;</li> <li>5. Escrita de testes para capturar requisitos;</li> <li>6. Uso de linguistic methods;</li> </ol> <p style="text-align: right;"><small>Cin.ufpe.br</small></p>
--	--

 <p>Também conhecido como <i>Grooming</i>.</p> <h3><b>Product Backlog Refinement</b></h3> <p style="text-align: right;"><small>Cin.ufpe.br</small></p>	 <h3><i>Product Backlog Refinement</i></h3> <ul style="list-style-type: none"> <li>• Qual objetivo?       <ul style="list-style-type: none"> <li>– Adicionar detalhes, estimativas e ordem aos itens de backlog, mas também pode ser útil para: remover incertezas, quebrar stories, identificar riscos, identificar dependências, etc.</li> </ul> </li> <li>• Quem deve participar?       <ul style="list-style-type: none"> <li>– Time de desenvolvimento* e o PO (Product Owner);</li> </ul> </li> <li>• Quando deve acontecer?       <ul style="list-style-type: none"> <li>– Sprint atual (meio ~ fim), mas focando em sprints futuras.</li> </ul> </li> </ul> <p style="text-align: right;"><small>Cin.ufpe.br</small></p>
---	---

 <h3><i>Product Backlog Refinement</i></h3> <ul style="list-style-type: none"> <li>• Alguns Benefícios:       <ul style="list-style-type: none"> <li>– Aumenta a eficiência do time reduzindo as incertezas;</li> <li>– Estimativas mais assertivas;</li> <li>– Atrasos devidos a dependências externas são descobertas cedo;</li> <li>– Reduz o tempo do planning;</li> </ul> </li> </ul> <p style="text-align: right;"><small>Cin.ufpe.br</small></p>	 <h3><i>Product Backlog Refinement</i></h3> <ul style="list-style-type: none"> <li>• Boas práticas:       <ul style="list-style-type: none"> <li>– Definir o objetivo da sessão: se possível enviar a lista de <i>stories</i> previamente. Os participantes devem sair da reunião com um entendimento claro do objetivo das próximas sprints;</li> <li>– Evento regular na agenda dos participantes (geralmente, alguns dias antes do planning);</li> </ul> </li> </ul> <p style="text-align: right;"><small>Cin.ufpe.br</small></p>
--	---



## Product Backlog Refinement

- Referências e Material Adicional:
  - [Scrum Guide](#);
  - [Introduction to Backlog Grooming - EasyAgile](#)
  - [Scrum.org vídeo no Youtube](#)
  - [Product Backlog Refinement por Mike Cohn](#)

©n.utpe.br



Cerimônia oficial do Scrum Guide. Também chamada de sprint demo.

## Sprint Review

©n.utpe.br



## Sprint Review

- Qual objetivo?
  - Apresentar o incremento com o intuito de obter feedback (aprovação ?) e promover a colaboração;
- Quem deve participar?
  - Time de desenvolvimento, PO (Product Owner), o *Scrum Master*, o gerente e outros *stakeholders*.
- Quando deve acontecer?
  - Ao final da *sprint*.

©n.utpe.br



## Sprint Review

- Alguns Benefícios:
  - Forma de aprendizado;
  - Engajamento dos *stakeholders*;
  - *Feedback* rápido;
  - *Team Building*;
  - Melhoria na auto estima do time;

©n.utpe.br



## Sprint Review

- Boas Práticas:
  - Usuários finais participam da review sempre que possível;
  - O foco da demo deve ser uma experiência real do usuário (não apenas demonstrar funcionalidade);
  - Preparar e revisar tudo antes da reunião;

©n.utpe.br



## Sprint Review

- Referências e Material Adicional:
  - [Scrum Guide](#)
  - [Scrum.org - Youtube vídeo](#)
  - [Artigo de Paetsch e outros](#)

©n.utpe.br



Participação do cliente na escrita ou revisão.

## Testes de Aceitação

On.utpa.br



## Testes de Aceitação

- Qual objetivo?
  - Descrever o comportamento do software na forma de um exemplo ou um cenário de uso (geralmente, criado das estórias de usuários);
- Quem deve escrever?
  - Idealmente, o cliente. No entanto, o time pode escrever desde que o cliente revise.
- Quando deve acontecer?
  - Antes do time de desenvolvimento começar a implementação.

On.utpa.br



## Testes de Aceitação

- Alguns Benefícios:
  - Contrato claro entre cliente e desenvolvedores (ajuda na definição de done);
  - Colaboração mais próxima do cliente;
  - Captura de requisitos do usuário (de forma direta);
  - Ser usado como documentação;

On.utpa.br



## Testes de Aceitação

- Boas Práticas:
  - Automatizar testes de aceitação;
  - Executar sempre;
  - Manter em mente o que realmente é teste de aceitação;

On.utpa.br



## Testes de Aceitação

- Alguns *frameworks*/ferramentas:
  - [Robot Framework](#);
  - [CodeceptJ](#);
  - [Calabash \(mobile apps\)](#);
  - [Gauge](#);
  - [Cucumber](#);
  - [Robotium](#) (mobile)

On.utpa.br

```
Feature: Is it Friday yet?
  Everybody wants to know when it's Friday

Scenario Outline: Today is or is not Friday
  Given today is <day>
  When I ask whether it's Friday yet
  Then I should be told <answer>

Examples:
| day | answer |
| "Friday" | "TGIF" |
| "Sunday" | "Nope" |
| "anything else!" | "Nope" |
```



## Testes de Aceitação

- Referências e Material Adicional:
  - [Acceptance Testing - Agile Alliance](#)
  - [Artigo do Haugset;](#)
  - [Artigo do Miler e outros;](#)

cin.ufpe.br



Requisito, teste e código

## Matrix de Rastreabilidade

cin.ufpe.br



## Matrix de Rastreabilidade

- Qual objetivo?
  - Obter e manter um link entre as suítes de teste e o código *under* testes.
- Quem deve escrever?
  - O time;
- Quando deve acontecer?
  - Durante o desenvolvimento ou quando houver alterações no código.

cin.ufpe.br



## Matrix de Rastreabilidade

- Alguns Benefícios:
  - Testes podem ser usados como documentação;
  - Muito útil on refatorações;
- Limitações:
  - Poucas ferramentas/plugins que otimizem o processo;

cin.ufpe.br



## Matrix de Rastreabilidade

- Referências e Material Adicional:
  - [Artigo do Qusef;](#)
  - [Artigo do Qusef e outros;](#)

cin.ufpe.br



Levantamento mais cedo possível.

## Cenários de Teste

cin.ufpe.br



## Cenários de Teste

- Qual objetivo?
  - Levantar os cenários de testes o mais cedo possível.
- Quem deve escrever?
  - Time (especialmente, QAs).
- Quando deve acontecer?
  - Idealmente, após a reunião de refinement e antes do planning.

Cin.ufpe.br



## Cenários de Teste

- Alguns Benefícios:
  - Antecipação de dependências/riscos;
  - Cenários *edges* serão conhecidos por todos (principalmente, a pessoa responsável pela implementação);
  - Durante a divisão de tasks, já ter um *brainstorming* inicial dos cenários;
  - Uso de *mind maps*;
  - Levar em consideração requisitos não funcionais;

Cin.ufpe.br



## Cenários de Teste

- Referências e Material Adicional:
  - [Helping Testers article](#);
  - [EuroStar Software Testing article](#);
  - [Segue Technologies article](#);

Cin.ufpe.br



Futuro. Artigo do [Goetz](#).

## Linguistic Methods

Cin.ufpe.br



Obrigada!  
Alguma **dúvida**?

Karla Silva - kmbs@cin.ufpe.br

Cin.ufpe.br

## APÊNDICE I – SLIDES DA APRESENTAÇÃO 5 - *FEEDBACK*

 <p><b>Projeto de Pesquisa</b> <b>CIn/Rec Tecnologia &amp; UFPE</b></p> <p>Programa de Pós-Graduação Karla Silva - kmbs@cin.ufpe.br</p> <p><small>Cin.ufpe.br</small></p>	 <p><b>Ações Escolhidas</b></p> <ol style="list-style-type: none"><li>1. <i>Product Backlog Refinement</i> (grooming);</li><li>2. <i>Sprint Review</i>;</li><li>3. Participação na escrita de testes de aceitação;</li><li>4. Rastreabilidade entre casos de testes e código;</li><li>5. Escrita de testes para capturar requisitos;</li><li>6. Uso de linguistic methods;</li></ol> <p><small>Cin.ufpe.br</small></p>
 <p><b>Feedback</b></p> <ol style="list-style-type: none"><li>1. Como foi a implementação das ações?</li><li>2. As ações implementadas ajudaram a entender melhor os requisitos?</li><li>3. Em resumo...<ol style="list-style-type: none"><li>a. O que foi bom?</li><li>b. O que podemos melhorar?</li><li>c. O que não funcionou de jeito nenhum?</li><li>d. Sentimentos?</li></ol></li></ol> <p><small>Cin.ufpe.br</small></p>	 <p><b>Obrigada!</b> <b>Alguma dúvida?</b></p> <p>Karla Silva - kmbs@cin.ufpe.br</p> <p><small>Cin.ufpe.br</small></p>