



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JARBELE CÁSSIA DA SILVA COUTINHO

AGILE ETEASY: UM MÉTODO PARA APLICAÇÃO DE TESTES
EXPLORATÓRIOS EM CONTEXTOS ÁGEIS

CAMPINA GRANDE - PB

2022

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Agile ETeasy: um método para aplicação de Testes
Exploratórios em contextos ágeis

Jarbele Cássia da Silva Coutinho

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutora em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Engenharia de Software

Dr. Wilkerson de Lucena Andrade

Dra. Patrícia Duarte de Lima Machado

(Orientação)

Campina Grande, Paraíba, Brasil

©Jarbele Cássia da Silva Coutinho, 08/08/2022

C871a

Coutinho, Jarbele Cássia da Silva.

Agile Eteasy: um método para aplicação de testes exploratórios em contextos ágeis / Renato Lima da Silva. - Campina Grande, 2022.
159 f. : il. color.

Tese (Doutorado em Engenharia Elétrica e Informática) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2022.

"Orientação: Prof. Dr. Wilkerson de Lucena Andrade, Prof. Dra. Patrícia Duarte de Lima Machado"

Referências.

1. Teste de Software. 2. Teste Exploratório. 3. Desenvolvimento Ágil. 4. Design Science Research. I. Andrade, Wilkerson de Lucena. II. Machado, Patrícia Duarte de Lima. III. Título.

CDU 004.4(043)



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO CIENCIAS DA COMPUTACAO
Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

JARBELE CÁSSIA DA SILVA COUTINHO

AGILE ETEASY: UM MÉTODO PARA APLICAÇÃO DE TESTES EXPLORATÓRIOS EM CONTEXTOS ÁGEIS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título Doutor em Ciência da Computação.

Aprovada em: 08/08/2022

Prof. Dr. WILKERSON DE LUCENA ANDRADE, Orientador, UFCG

Profa. Dra. PATRICIA DUARTE DE LIMA MACHADO, Orientadora, UFCG

Prof. Dr. EVERTON LEANDRO GALDINO ALVES, Examinador Interno, UFCG

Profa. Dra. ELIANE CRISTINA DE ARAÚJO, Examinadora Interna, UFCG

Profa. Dra. CARLA TACIANA LIMA LOURENCO SILVA SCHUENEMANN, Examinadora Externa, UFPE

Prof. Dr. VALDIVINO ALEXANDRE DE SANTIAGO JÚNIOR, Examinador Externo, INPE



Documento assinado eletronicamente por **WILKERSON DE LUCENA ANDRADE, PROFESSOR DO MAGISTERIO SUPERIOR**, em 09/08/2022, às 13:57, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **Carla Taciana Lima Lourenço Silva Schuenemann, Usuário Externo**, em 09/08/2022, às 15:11, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **EVERTON LEANDRO GALDINO ALVES, PROFESSOR 3 GRAU**, em 09/08/2022, às 15:46, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **ELIANE CRISTINA DE ARAUJO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 09/08/2022, às 20:31, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **PATRICIA DUARTE DE LIMA MACHADO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 10/08/2022, às 07:42, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **Valdivino Alexandre de Santiago Júnior, Usuário Externo**, em 11/08/2022, às 09:00, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufcg.edu.br/autenticidade>, informando o código verificador **2612233** e o código CRC **1682FDCC**.

Resumo

A inserção de práticas ágeis no desenvolvimento de *software* aumentou exponencialmente, frente as demandas do mercado quanto à agilidade e alta qualidade dos produtos de *software* desenvolvidos. Assim, os métodos ágeis têm estimulado, nas equipes, práticas mais simples, dinâmicas e iterativas que consideram a redução de artefatos, a comunicação frequente entre os *stakeholders*, a flexibilidade para mudanças, dentre outros aspectos. Um estilo de teste que tem se difundido no ambiente ágil é o Teste Exploratório (ET), pois permite que os profissionais aprendam rapidamente, ajustem seus testes e, nesse processo, encontrem problemas de *software* que muitas vezes não são previstos em planos de testes. Entretanto, as abordagens de ET evidenciadas pela literatura não são aplicáveis no contexto prático de equipes ágeis. Outros motivos se dão pela ausência de um apoio mais específico para o planejamento e execução do ET, considerando o perfil multifuncional da equipe; os artefatos de ET estão obsoletos para o contexto atual das equipes; a abordagem *Session-Based Test Management* (SBTM), por exemplo, não se aplica ao contexto prático; dificuldade em reproduzir os ET, devido à ausência de artefatos robustos; dentre outros. Neste sentido, o objetivo desta pesquisa é apoiar a implementação de Testes Exploratórios em ambientes de desenvolvimento de *software* que adotam práticas de métodos ágeis. Para isso, utilizamos um método de pesquisa orientado à resolução de problemas, conhecido como *Design Science Research* (DSR). Com base em um rigor metodológico, foram aplicados dois ciclos da DSR. Os resultados gerados nesta pesquisa indicam que o método *Agile ETeasy* é de fácil entendimento e de fácil uso por equipes ágeis que atuam em um contexto de documentação informal e mudanças frequentes nos requisitos. Diante do que foi investigado, podemos concluir que o artefato proposto com a DSR é viável de ser utilizado na prática por equipes ágeis, pois possibilita a mitigação das principais limitações da aplicação de ET em projetos que adotam métodos ágeis; e, a formalização prática de ET, que consideram o contexto multifuncional das equipes ágeis, e que fornece resultados rápidos, com pouca quantidade de recursos.

Abstract

The insertion of agile practices in the development of software has increased exponentially, in the face of market demands regarding the agility and high quality of the software products developed. Thus, agile methods have stimulated, in teams, simpler, dynamic, and iterative practices that consider the reduction of artifacts, frequent communication between stakeholders, and flexibility for changes, among other aspects. A style of testing that has become widespread in the agile environment is Exploratory Testing (ET), as it allows professionals to learn quickly, adjust their tests and, in the process, find software problems that are often not foreseen in test plans. However, the ET approaches evidenced by the literature are not applicable in the practical context of agile teams. Other reasons are given by the absence of more specific support for the planning and execution of the ET, considering the multi-functional profile of the team; ET artifacts are obsolete for the current context of teams; the Session-Based Test Management (SBTM), for example, does not apply to the practical context; difficulty in reproducing the ET, due to the absence of powerful artifacts; among others. In this sense, the objective of this research is to support the implementation of ET in software development environments that adopt agile methods practices. For this, we use a problem-oriented research method, known as Design Science Research (DSR). Based on methodological rigor, two DSR cycles were applied. The results generated in this research indicate that the Agile ETeasy method is easy to understand and use by agile teams that work in the context of informal documentation and frequent changes in requirements. In view of what was investigated, we can conclude that the proposed artifact with the DSR is viable to be used in practice by agile teams, as it enables the mitigation of the main limitations of the application of ET in projects that adopt agile methods; and, the practical formalization of ET, which consider the cross-functional context of agile teams, and which provides quick results, with a little number of resources.

Dedicatória

Dedico esta tese aos meus pais, Cosme Antonio e Josefa Teixeira: meu porto seguro!

Agradecimentos

Ao longo destes 5 anos e meio de doutorado, muitas situações pessoais aconteceram que me fizeram pensar em desistir do doutorado, diversas vezes. E se, hoje, entrego esta tese é porque muitos me apoiaram para que eu a concluísse.

Durante o doutorado, fui aprovada em concurso público federal, para o cargo de professora do Magistério Superior na Universidade Federal Rural do Semi-Árido (UFERSA), campus Pau dos Ferros. Precisei mudar de cidade. E, continuei conciliando os estudos com o trabalho. Depois de um tempo me afastei integralmente das atividades para me dedicar à tese. Após 3 meses de afastamento, teve início a pandemia do *CORONA VIRUS DISEASE* (Doença do Coronavírus) ou *SARS-CoV-2*, que perdura até os dias atuais. Durante esta pandemia, meu esposo (Alexsandro) foi diagnosticado com um *Linfoma de Hodgkin com Predominância Linfócita Nodular*, um tumor raro. Eu engravidei do meu primeiro filho. Durante a gravidez, iniciamos e encerramos o tratamento quimioterápico do meu esposo. Ele foi curado! Depois disso, eu perdi meu tio (Damião) para a Covid-19. E, em seguida, perdi uma das minhas irmãs (Maria Jaberlánye) também para a Covid-19. Essas perdas foram avassaladoras. Mas, seguimos. Meses depois, meu bebê nasceu, em meio a complicações no final da gestação. E, prossegui a tese na companhia de Theo, meu primeiro filho.

Porque estou fugindo do protocolo e relatando isso nos Agradecimentos da minha tese? A resposta é clara: se não fosse o apoio dos meus orientadores, meus familiares e alguns colegas, eu não teria conseguido. Por isso, aqui, eu faço questão de agradecer!

Primeiramente, a Deus por ter cuidado de mim em cada dificuldade pessoal, que impactou diretamente em minha pesquisa de doutorado. Por ter me dado a saúde psicológica, física e emocional necessárias para prosseguir. A Nossa Senhora da Conceição, a quem sou devota, que intercedeu e cuidou de mim em cada situação vivida nos últimos anos.

Aos meus orientadores, Prof. Dr. Wilkerson Andrade e Profa Patrícia Machado, muito obrigada! Costumo dizer que não tenho orientadores, mas "anjos" que me acolheram em cada dificuldade, com compreensão e sabedoria. Professores, sem o apoio de vocês, eu não teria conseguido! Minha gratidão e admiração pelos "profissionais-humanos" que são, e em quem me espelho em minha carreira profissional. Obrigada por toda dedicação em minha orientação.

Ao meu esposo, Alessandro Coutinho, que me acompanhou diariamente em cada batalha, compartilhando as dificuldades, as dúvidas e as conquistas durante o doutorado. Sua compreensão e companheirismo foram fundamentais em minha jornada!

Ao meu filho, Theo Gustavo, que desde o seu nascimento tem me ensinado sobre ser uma pessoa melhor, mais decidida e mais forte. Esta tese é por você, meu filho!

Aos meus pais, Cosme e Zefinha, minha admiração e meu amor! Agradeço todo o suporte, conselhos e compreensão, principalmente nas (inúmeras) vezes em que precisei me ausentar de momentos com vocês por conta do doutorado. A filha do agricultor e da professora de ensino básico, só tem orgulho de vocês!

Às minhas irmãs, Jaberlânje (*in memorian*), Jacimara e Simone, obrigada por todo suporte e por “me ajudarem a respirar” durante os períodos mais intensos da pesquisa! Durante este doutorado, vocês sempre foram meu equilíbrio! Aos meus amados sobrinhos, Ian e Esther: espero ser exemplo para vocês, meus pequenos.

À Universidade Federal Rural do Semi-Árido (UFERSA), campus Pau dos Ferros-RN, por contribuir com minha qualificação profissional através do período (um ano e nove meses) em que estive afastada das atividades institucionais para me dedicar ao doutorado. Obrigada!

Às profissionais do *Software Practices Laboratory* (SPlab) da Universidade Federal de Campina Grande (UFCG) por contribuírem com a realização da minha pesquisa. Em especial, quero agradecer à Larissa, Dalton e Raphael, pela troca de conhecimentos durante o doutorado.

Às equipes de desenvolvimento de *software* e a todos os participantes do SPlab/UFCG que dispuseram de seu tempo para contribuir com os resultados gerados nesta tese.

À todos os integrantes do Programa de Pós-Graduação em Ciência da Computação (PPgCC) da UFCG por todo suporte necessário durante o doutorado.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por ter custeado os primeiros meses do meu doutorado através de bolsas de incentivo à pesquisa.

Aos professores membros desta banca examinadora, Prof. Dr. Everton, Profa. Dra. Eliane, Profa. Dra. Carla e Prof. Dr. Valdivino, agradeço as contribuições fornecidas à tese.

E, a todos que direta ou indiretamente contribuíram com minha formação profissional durante este doutorado.

*“Meu propósito de vida é ensinar
tudo aquilo que sei. Transformar
vidas através do conhecimento.”*

Maria Jaberlânye da Silva Nelo

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Contexto e Motivação | 1 |
| 1.2 | Definição do Problema | 3 |
| 1.3 | Objetivos da Pesquisa | 4 |
| 1.4 | Questões de Pesquisa | 4 |
| 1.5 | Metodologia adotada | 5 |
| 1.6 | Estrutura da Tese | 6 |
| 2 | Contextualização | 7 |
| 2.1 | Métodos ágeis em Engenharia de <i>Software</i> | 7 |
| 2.2 | Engenharia de Requisitos Ágil | 9 |
| 2.3 | Teste de <i>Software</i> no desenvolvimento de <i>software</i> ágil | 11 |
| 2.4 | Testes Exploratórios | 14 |
| 2.4.1 | <i>Session Based Test Management</i> (SBTM) | 15 |
| 2.4.2 | <i>Exploratory Testing in Pairs</i> | 17 |
| 2.4.3 | Team Exploratory Testing Sessions (TET) | 17 |
| 2.4.4 | <i>Tours</i> | 18 |
| 2.4.5 | <i>Rule-Based Exploratory Testing</i> | 19 |
| 2.5 | <i>Design Science Research</i> | 20 |
| 2.6 | Considerações Finais do Capítulo | 23 |
| 3 | O Método Agile <i>ETeasy</i> | 24 |
| 3.1 | Método Agile <i>ETeasy</i> : Etapa de Planejamento | 24 |
| 3.2 | Método Agile <i>ETeasy</i> : Etapa de Execução | 27 |

| | | |
|----------|---|------------|
| 3.3 | Método <i>Agile ETeasy</i> : Etapa de Encerramento | 28 |
| 3.4 | Orientações Adicionais do Método <i>Agile ETeasy</i> | 32 |
| 3.5 | Considerações finais do capítulo | 34 |
| 4 | Aplicação da <i>Design Science Research</i> | 35 |
| 4.1 | Metodologia da DSR | 35 |
| 4.1.1 | Metodologia aplicada no Ciclo 1 | 37 |
| 4.1.2 | Metodologia aplicada no Ciclo 2 | 41 |
| 4.2 | Ciclo 1 da DSR | 44 |
| 4.2.1 | Investigação do Problema de Pesquisa | 44 |
| 4.2.2 | <i>Design</i> da Solução | 59 |
| 4.2.3 | Validação da Solução | 68 |
| 4.2.4 | Implementação da Solução | 70 |
| 4.3 | Ciclo 2 da DSR | 73 |
| 4.3.1 | Avaliação da Implementação | 73 |
| 4.3.2 | <i>Design</i> da Solução | 77 |
| 4.3.3 | Validação da Solução | 79 |
| 4.3.4 | Implementação e Avaliação da Solução | 84 |
| 4.4 | Considerações Finais do Capítulo | 92 |
| 5 | Trabalhos Relacionados | 93 |
| 5.1 | Testes Exploratórios no contexto ágil | 93 |
| 5.1.1 | Suporte à execução de ET na indústria ágil | 95 |
| 5.2 | Posicionamento desta pesquisa em relação aos trabalhos relacionados | 98 |
| 5.3 | Considerações finais do Capítulo | 103 |
| 6 | Considerações Finais | 104 |
| 6.1 | Conclusões | 104 |
| 6.2 | Contribuições | 108 |
| 6.3 | Ameaças a validade | 109 |
| 6.4 | Limitações | 111 |
| 6.4.1 | Limitações da implementação do Ciclo 1 da DSR | 111 |

| | | |
|----------|--|------------|
| 6.4.2 | Limitações da implementação do Ciclo 2 da DSR | 112 |
| 6.5 | Trabalhos Futuros | 112 |
| A | Apêndice A - Protocolo da Revisão Sistemática da Literatura (RSL) | 123 |
| A.1 | Questões de Pesquisa investigadas | 123 |
| A.2 | Busca por trabalhos primários | 124 |
| A.3 | CrITÉrios de Inclusão e Exclusão | 125 |
| A.4 | CrITÉrios de Qualidade | 125 |
| A.5 | Seleção dos Trabalhos | 126 |
| B | Apêndice B - Ferramentas para Teste Exploratório | 128 |
| C | Apêndice C - Entrevista realizada na <i>Investigação do Problema</i> do Ciclo 1 | 132 |
| D | Apêndice D - Protocolo do Experimento sobre SBTM | 134 |
| D.1 | Visão Geral | 134 |
| D.2 | Planejamento do Experimento | 135 |
| D.2.1 | Planejamento do Curso de ET | 136 |
| D.2.2 | Planejamento do <i>Brainstorming</i> | 138 |
| D.3 | Execução do Experimento | 140 |
| D.3.1 | Execução do Curso | 140 |
| D.3.2 | Execução do <i>Brainstorming</i> | 141 |
| D.4 | Procedimentos de Análise | 142 |
| E | Apêndice E - Questionário aplicado na <i>Validação da Solução</i> no Ciclo 1 | 143 |
| F | Apêndice F - Questionário aplicado na <i>Validação da Solução</i> no Ciclo 2 | 146 |
| G | Apêndice G - Protocolo do Estudo de Caso da etapa de <i>Implementação da Solução</i> do Ciclo 1 e 2 | 149 |
| G.1 | Visão Geral | 149 |
| G.2 | <i>Design</i> do Estudo de Caso | 150 |
| G.3 | Preparação para coleta de dados | 150 |
| G.4 | Coleta de Dados | 151 |

| | | |
|----------|---|------------|
| G.5 | Análise dos dados coletados | 152 |
| H | Apêndice H - Questionário inicial da <i>Implementação da Solução</i> do Ciclo 1 e 2 | 153 |
| I | Apêndice I - Diário de Bordo usado na <i>Avaliação da Implementação</i> do Ciclo 1 e 2 | 155 |
| J | Apêndice J - Entrevista realizada na <i>Avaliação da Implementação</i> do Ciclo 1 e 2 | 156 |
| K | Apêndice K - Método <i>Agile ETeasy</i>: Artefato de Planejamento | 158 |
| L | Apêndice L - Método <i>Agile ETeasy</i>: Artefato de Encerramento | 159 |

Lista de Símbolos

AP - Atividade Prática

ASD - Agile Software Development

ATDD - Acceptance Test-Driven Development

DSDM - Dynamic System Development Method

DSR - Design Science Research

ET - Exploratory Testing

E2E - End-to-End

FDD - Feature-Driven Development

GUI - Graphical User Interface

LETT - LEET Enhances Exploratory Testing

OM - Opportunity Map

PMEs - Pequenas e Médias Empresas de software

PMI - Project Management Institute

QE - Questão Específica

QP - Questões de Pesquisa

R-BET - Rule-Based Exploratory Testing

RE - Requirements Engineering

RSL - Revisão Sistemática da Literatura

Sars-CoV-2 - Corona Virus Disease

SAST - Symposium on Systematic Automated Software Testing

SBES - Brazilian Symposium on Software Engineering

SBTM - Session-Based Test Management

SPLAB - Software Practices Laboratory

SRS - Software Requirements Specifications

SUT - *System Under Test*

ST - *Software Testing*

TCT - *Testes com Casos de Teste*

TDD - *Test Driven Development*

TET - *Team Exploratory Testing Sessions*

UFCG - *Universidade Federal de Campina Grande*

XP - *Extreme Programming*

Lista de Figuras

| | | |
|-----|--|-----|
| 2.1 | Quadrantes para Testes Ágeis (Adaptado de Crispin and Gregory [2][1][26]). | 12 |
| 2.2 | Ciclo Regulador da DSR, proposto por Wieringa [2][1][78]. | 21 |
| 2.3 | Instruções gerais para a condução da <i>Design Science Research</i> (Traduzido de Dresch, Lacerda e Jr. [2][1][31]). | 22 |
| 3.1 | Etapa de Planejamento do método <i>Agile ETeasy</i> | 25 |
| 3.2 | Etapa de Execução do método <i>Agile ETeasy</i> | 27 |
| 3.3 | Etapa de Encerramento do método <i>Agile ETeasy</i> | 29 |
| 4.1 | Ciclos da DSR aplicados nesta pesquisa. | 36 |
| 4.2 | Execução do <i>Brainstorming</i> | 60 |
| 4.3 | Estrutura geral do Método <i>Agile ETeasy</i> | 65 |
| 4.4 | Primeira versão dos artefatos de teste do método <i>Agile ETeasy</i> | 67 |
| 4.5 | <i>Print Screen</i> do <i>website</i> do método <i>Agile ETeasy</i> | 67 |
| A.1 | Etapas seguidas nesta RSL. | 127 |
| D.1 | Fases de desenvolvimento deste experimento. | 135 |

Lista de Tabelas

| | | |
|-----|---|-----|
| 3.1 | Atividade 1. Definir escopo de teste. | 26 |
| 3.2 | Atividade 2. Aplicar os testes. | 28 |
| 3.3 | Atividade 3. Observar os Resultados. | 28 |
| 3.4 | Atividade 4. Registrar o teste. | 29 |
| 3.5 | Atividade 5. Alinhar as melhorias. | 31 |
| 4.1 | Análise das ferramentas em relação a disponibilidade de acesso e suporte à plataforma. | 50 |
| 4.2 | Análise das ferramentas em relação às funcionalidades. | 51 |
| 4.3 | Tempo de duração de cada entrevista | 53 |
| 4.4 | Caracterização dos entrevistados. | 53 |
| 4.5 | Depoimentos dos entrevistados sobre as dificuldades em executar testes nos projetos. | 55 |
| 4.6 | Estrutura do Curso de Testes Exploratórios usando a abordagem SBTM . . . | 58 |
| 4.7 | Síntese dos registros no Diário de Bordo. | 75 |
| 5.1 | Síntese dos trabalhos relacionados à esta pesquisa. | 101 |
| 5.2 | Análise comparativa das abordagens destacadas nos trabalhos relacionados. | 102 |
| B.1 | Endereços <i>web</i> de acesso às ferramentas selecionadas. | 129 |
| D.1 | Perguntas específicas do <i>brainstorming</i> | 139 |
| I.1 | Registro da experiência de implementação do Método <i>Agile ETeasy</i> | 155 |

Capítulo 1

Introdução

Este capítulo apresenta uma visão geral do trabalho. Assim, uma breve contextualização sobre Teste de *Software* e Teste Exploratório no contexto de métodos ágeis, bem como a motivação da pesquisa são discutidas na Seção 1.1. Depois disso, discutimos o problema de pesquisa na Seção 1.2. Na sequência, os objetivos do estudo e as questões de pesquisa são expostos, respectivamente, nas Seções 1.3 e 1.4. Apresentamos a metodologia de estudo adotada nesta tese, na Seção 1.5. Por fim, a estrutura geral da tese é apresentada na Seção 1.6.

1.1 Contexto e Motivação

Na Engenharia de *Software*, a atividade de Teste de *Software* (*Software Testing* - ST) ajudam a garantir a qualidade do *software*. Por meio desta atividade é possível detectar e solucionar erros que possam interferir no correto funcionamento do *software*. Portanto, Teste de *Software* é definido como a atividade que visa verificar se o *software* produzido foi implementado corretamente, ou seja, se atende aos requisitos especificados pelo cliente e atende suas expectativas [12; 16]. De forma complementar, Barbieri [7] destaca que o ST também visa verificar e validar os artefatos gerados durante seu processo de desenvolvimento.

Por meio de uma variedade de estratégias, técnicas e atores, o teste é amplamente utilizado na indústria de *software* para garantir a qualidade [12]. No entanto, o ST apresenta desafios complexos - como planejamento da atividade de teste, falta de profissionais especializados, dificuldades na implementação de um processo de teste, entre outros [42;

24] - tornando cada vez mais difícil garantir os níveis esperados de qualidade e confiabilidade em *software*.

No contexto ágil, identificar defeitos não é o único objetivo da atividade de Teste de *Software*. Para Crispin e Gregory [26], Teste de *Software* Ágil (*Agile Software Testing - Agile ST*) consiste em procurar continuamente maneiras da equipe fazer seu melhor trabalho para entregar *software* de qualidade. Para isso, a equipe pode utilizar diferentes estratégias para avaliar e melhorar a qualidade dos produtos de *software* desenvolvidos.

O *Agile ST* define um conjunto de práticas que incorpora técnicas de teste comumente usadas e considera valores ágeis para fazê-lo [26]. Para Maia et al. [54], no *Agile ST*, a equipe não está apenas envolvida na identificação de falhas, mas também em preveni-las. Dessa forma, o *Agile ST* é um desafio para testadores acostumados a usar métodos tradicionais, principalmente porque precisam ser proativos e iniciar os testes desde o início do projeto junto com os desenvolvedores [54].

Alguns estudos [4; 17; 74] têm investigado o *Agile ST* e indicado: 1) ênfase no uso de estratégias de desenvolvimento orientadas a testes unitários e de integração e testes de aceitação com o cliente, bem como a preocupação com estudos experimentais que possam medir os benefícios e dificuldades de usar tais estratégias; 2) a necessidade de ferramentas de teste, no contexto de métodos ágeis; 3) os principais fatores técnicos, culturais e gerenciais que impulsionam os testes em ambientes DevOps; 4) fatores que limitam a adoção do *Test Driven Development* (TDD) pela indústria; 5) entre outros.

Dentre as práticas de *Agile ST* existentes, têm-se os Testes Exploratórios (*Exploratory Testing - ET*). Este tipo de Teste de *Software* consiste em um processo em que o testador aprende, elabora o *design* do teste e executa simultaneamente enquanto explora o produto [6], pois não há planejamento de *scripts* ou a definição de casos de testes, definidos em planos de testes [40].

Em termos leigos, pode-se dizer que o ET permite que os profissionais aprendam rapidamente, ajustem seus testes e, nesse processo, encontrem problemas de *software* que muitas vezes não são previstos em planos ou *scripts* de teste [23]. Assim, ET é apropriado para identificar defeitos rapidamente, ou quando o projeto não possui uma documentação de requisitos, ou esta não é bem definida ou está desatualizada.

No geral, para apoiar o gerenciamento e a medição dos ET algumas abordagens são

destacadas pela literatura, como a abordagem *Session-Based Test Management (SBTM)*, *Checklists to Support Test Charter*, *Exploratory Testing in Pairs*, *Team Exploratory Testing Sessions - (TET)*, *Tours*, *Rule-Based Exploratory Testing* [33; 36; 71], dentre outras.

Desse modo, assim como expandir os métodos ágeis trouxe benefícios para as equipes, também gerou novos desafios na implementação prática de alguns tipos de testes, como o ET. Algumas abordagens de ET, quando inseridas no contexto ágil, não se adequam completamente às necessidades das equipes.

1.2 Definição do Problema

A qualidade do *software* é um dos fatores determinantes para o sucesso de qualquer projeto. No entanto, garantir isso não é uma tarefa trivial. Hoje, as equipes de desenvolvimento de *software* têm focado suas práticas no desenvolvimento de produtos que atendam às necessidades dos clientes para atender às crescentes demandas por mudanças de flexibilidade e pressões de prazos cada vez mais curtos [24].

Assim, a utilização de métodos ágeis no desenvolvimento de *software* têm sido uma alternativa comum adotada pelas empresas, devido à facilidade de adaptação às mudanças e à adoção de processos mais simplificados. No entanto, adotar tais abordagens implica em adaptar o contexto de desenvolvimento de todas as fases do projeto [69]. Isso também inclui as atividades de ST. O problema é que as estratégias validadas com equipes ágeis e que viabilizam a execução de testes neste cenário, apresentam algumas limitações. Principalmente, no que se refere à execução dos Testes Exploratórios [33; 35; 55; 62].

Embora algumas abordagens tenham sido desenvolvidas e se mostrado eficientes na aplicação do ET, como o *Session-Based Test Management (SBTM)*, *Checklists to Support Test Charter*, *Exploratory Testing in Pairs*, *Team Exploratory Testing Sessions - (TET)*, *Tours* e *Rule-Based Exploratory Testing* [33; 36; 71], algumas limitações são percebidas quando aplicadas no contexto de equipes ágeis, como pouco tempo disponível para executar o teste, processos muito teóricos para serem aplicados e artefatos incompatíveis com a rotina de trabalho [23].

Alguns estudos discutem formas de tornar mais prática a aplicação de ET em projetos

ágeis [9; 10; 36; 53; 65; 81]. Entretanto, os resultados mostram que os desafios ainda são recentes e indicam para a necessidade de uma abordagem bem definida e aplicável ao contexto prático e multifuncional das equipes ágeis.

1.3 Objetivos da Pesquisa

Considerando o problema de pesquisa exposto na Seção 1.2, o objetivo principal deste trabalho é apoiar a implementação de Testes Exploratórios em ambientes de desenvolvimento de *software* que adotam práticas de métodos ágeis. Assim, para atingir o objetivo principal, foram definidos três objetivos específicos:

1. Mapear informações sobre abordagens, práticas, ferramentas e artefatos existentes que apoiam a aplicação prática de Testes Exploratórios desenvolvidos no contexto ágil;
2. Investigar as limitações de aplicar ET no contexto ágil;
3. Elaborar e avaliar uma estratégia de ET que considere o cenário de desenvolvimento ágil;

1.4 Questões de Pesquisa

Tendo em vista os objetivos de pesquisa apresentados na Seção 1.3, a principal Questão de Pesquisa (QP) desta tese é: *O que fazer para conseguir integrar Testes Exploratórios, como prática de teste, no cotidiano de equipes ágeis?*. Para entender melhor o contexto desse questionamento, esta pesquisa de doutorado se concentra nas seguintes Questões Específicas (QE), em um escopo mais específico de Teste de *Software* Ágil: o Teste Exploratório.

QE1. *Quais são as principais abordagens, práticas, ferramentas e artefatos existentes que podem auxiliar os ET aplicados no contexto ágil?*

QE2. *Quais são as principais limitações de aplicar ET no contexto ágil?*

QE3. *Como organizar a maneira de conduzir ET de modo a otimizar sua prática no contexto ágil?*

1.5 Metodologia adotada

Nestas circunstâncias garantir a relevância e o rigor de uma solução aplicável ao contexto citado é importante para a implantação efetiva dos ET nos projetos desenvolvidos por equipes ágeis. Para isso, esta pesquisa de doutorado é realizada por meio da metodologia de *Design Science Research* (DSR) [31; 41; 78].

A DSR apresenta-se como um método de pesquisa orientado à resolução de problemas, que visa apoiar o projeto e a avaliação de artefatos ou soluções com base no rigor metodológico [78]. Deste modo, a DSR é baseada na condução das seguintes etapas: Investigação do Problema; *Design* da Solução; Validação da Solução; Implementação da Solução; e, Avaliação da Implementação. Essas etapas encadeiam um ciclo, chamado por Wieringa et al. [78] de “Ciclo Regulador”. Este ciclo pode ser repetido quantas vezes forem necessárias até o refinamento completo do artefato, ou seja, até que o problema encontrado na etapa de Investigação seja solucionado.

Nesta tese, aplicamos dois ciclos da DSR para projetar, refinar e acompanhar a implementação da solução proposta em projetos reais desenvolvidos por profissionais de equipes ágeis. Na etapa de *Investigação do Problema*, foram realizadas (i) uma pesquisa bibliográfica sobre ET no contexto de desenvolvimento ágil, (ii) pesquisa por ferramentas de ET e (iii) entrevistas com profissionais de equipes ágeis, que culminaram na realização de (iv) um curso sobre ET com profissionais de equipes ágeis [23].

Na etapa de *Design da Solução* acompanhamos a execução de Sessões de ET em projetos reais desenvolvidos por equipes ágeis, conduzimos entrevistas com essas equipes e iniciamos a construção do método *Agile ETeasy*. Na etapa de *Validação da Solução* foi conduzido um estudo de validação do método *Agile ETeasy* por profissionais especialistas da área através da análise do método e da coleta de dados por meio de um questionário *online*. Na etapa de *Implementação da Solução* aplicamos um Estudo de Caso com uma equipe ágil, e avaliamos os resultados na etapa de *Avaliação da Implementação*, por meio de entrevistas, de um diário de bordo, e de um questionário *online* preenchido pelos participantes ao final do Estudo de Caso.

O segundo ciclo foi aplicado a partir das etapas de *Design da Solução*, com o refinamento do método; de *Validação da Solução*, com uma segunda validação feita com novos profissi-

onais especialistas da área; de *Implementação da Solução*, por meio de um segundo Estudo de Caso com uma equipe ágil; e, *Avaliação da Implementação*, por meio da análise dos resultados coletados e observados no Estudo de Caso anterior.

É importante destacar que esta pesquisa de doutorado está inserida no contexto do *Software Practices Laboratory* (SPLab) da Universidade Federal de Campina Grande (UFCG), e baseia-se em problemas reais de projetos de *software* desenvolvidos no SPLab. Deste modo, as etapas que compõem o primeiro e o segundo ciclo da DSR foram realizadas com profissionais que integram equipes ágeis de projetos de *software* do SPLab, com o intuito de adquirirmos novos *insights* para o método proposto.

1.6 Estrutura da Tese

Os demais capítulos que compõem este documento foram organizados para facilitar a compreensão do contexto e dos procedimentos adotados na condução da pesquisa. Assim, esta tese de doutorado é composta por sete capítulos.

Neste Capítulo, o contexto e a motivação de nossa pesquisa são apresentados. O Capítulo 2 expõe definições gerais que contextualizam a área de pesquisa investigada nesta tese e que fornecem um embasamento teórico ao leitor sobre Teste de *Software* e Teste Exploratório no contexto de métodos ágeis, bem como sobre *Design Science Research* (DSR).

O Capítulo 3 apresenta o método *Agile ETeasy*, destacando os elementos que o compõem e orientações para sua implementação em projetos reais.

O Capítulo 4 descreve os procedimentos metodológicos aplicados à pesquisa, por meio do planejamento das etapas da *Design Science Research*, da coleta e da análise de dados. E, apresenta os resultados de cada etapa da DSR.

O Capítulo 5 destaca e discute os principais trabalhos relacionados à Teste Exploratório na indústria ágil, com foco nas principais abordagens investigadas na literatura.

O Capítulo 6 apresenta as conclusões referentes aos objetivos alcançados, bem como as contribuições e as limitações da pesquisa, e as propostas de trabalhos futuros.

Capítulo 2

Contextualização

Neste capítulo é apresentada uma visão geral dos conceitos necessários para fundamentar as discussões desta pesquisa. Desse modo, são expostos os princípios do Método Ágil na Engenharia de *Software*, na Seção 2.1. Posteriormente, são descritas as principais definições sobre Engenharia de Requisitos no contexto Ágil (*Agile Requirements Engineering - Agile RE*), na Seção 2.2. Em seguida, apresentam-se as definições sobre Teste de *Software* no desenvolvimento ágil (*Agile Software Testing - Agile ST*) e Testes Exploratórios (*Exploratory Testing - ET*), respectivamente, na Seção 2.3 e na Seção 2.4. Por fim, a definição da *Design Science Research* - usada como procedimento metodológico nesta pesquisa - é apresentada na Seção 2.5 e, as conclusões deste capítulo são discutidas na Seção 2.6.

2.1 Métodos ágeis em Engenharia de *Software*

A partir da década de 1990 surgiram métodos (como *Crystal Methods*, SCRUM, XP, FDD, DSDM, dentre outros) que visam facilitar a forma como os *softwares* têm sido desenvolvidos. Em 2001, para estabelecer bases comuns entre esses métodos foi criada a *Agile Alliance* por meio do Manifesto Ágil [1].

O Manifesto Ágil defende que, para um método ser caracterizado como ágil, deve estar de acordo com os quatro valores e os doze princípios apontados no Manifesto [66]. Assim, o Manifesto Ágil considera que em métodos ágeis deve ser dada prioridade à quatro valores básicos [1]: (i) indivíduos e interações mais que processos e ferramentas; (ii) *software* em funcionamento mais que documentação abrangente; (iii) colaboração com o cliente mais que

negociação de contratos; e, (iv) responder a mudanças mais que seguir um plano.

Seguir esses valores implica em incorporar uma maior flexibilidade nos projetos, considerando as mudanças que ocorrem continuamente em seu escopo. Em métodos ágeis, o cliente trabalha em estreita colaboração com a equipe de desenvolvimento para alcançar essas definições e para validar, constantemente, o produto a ser entregue [66; 42]. Desse modo, os métodos ágeis priorizam a comunicação contínua e direta com o cliente em vez de uma documentação abrangente, visando a entrega rápida do *software* operacional [14]. Para isso, alguns métodos adotam uma estratégia de trabalho incremental e iterativa [51].

Portanto, considerando que o método ágil abrange vários modelos, são descritos, a seguir, alguns dos principais métodos que visam apoiar o desenvolvimento de *software*.

1. *Crystal Methods*: Consiste em uma família de metodologias adequadas para diferentes tipos de projeto, que variam de acordo com o tamanho da equipe e nos riscos envolvidos no projeto. Para caracterizar o projeto, são verificados o tamanho da equipe (definida em cores: claro, amarelo, laranja, vermelho e azul), a criticidade e a prioridade do projeto. Em síntese, o *Crystal* tem sete propriedades: entrega frequente, melhoria reflexiva, comunicação osmótica, segurança pessoal, foco, fácil acesso a usuários experientes e requisitos para o ambiente técnico [18].
2. *SCRUM*: É um *framework* para desenvolver e manter produtos complexos. Consiste em um método ágil para gestão e planejamento de projetos de *software*. Os projetos são divididos em ciclos, chamados de *Sprint* - iniciando com um planejamento e encerrando com uma revisão. As funcionalidades a serem implementadas são mantidas em *Product Backlog*. Em uma reunião de planejamento (*Sprint Planning Meeting*), o *Product Owner* prioriza os itens do *Product Backlog* e a equipe define a *Sprint*. Cada dia de *Sprint*, há uma breve reunião (*Daily Scrum*) e ao final da *Sprint* faz-se uma *Sprint Retrospective* [63].
3. *eXtreme Programming (XP)*: Busca ajudar as equipes a desenvolver *software* de alta qualidade em um cenário de requisitos vagos e que mudam rapidamente. Para isso, concentra-se em doze práticas: jogo de planejamento, entregas frequentes, uso de metáforas, projeto simplificado, testes, refatoração, programação em pares, propriedade

coletiva, integração contínua, ritmo sustentável (40 horas semanais), cliente presente, padrões de codificação [11].

4. *Feature-Driven Development* (FDD): Busca entregar aplicabilidade, ou seja, o desenvolvimento é orientado à funcionalidade. Para isso, possui cinco processos básicos: desenvolver um modelo geral, criar uma lista de funcionalidades, planejar, modelar e construir por funcionalidade. Em síntese, o FDD buscar atuar no processo de desenvolvimento [61].
5. *Dynamic System Development Method* (DSDM): É útil para projetos que necessitam de uma execução rápida e possuem requisitos que mudam constantemente. Para isso, o DSDM consiste em três fases sequenciais: pré-projeto, ciclo de vida e pós-projeto. Os princípios que fundamentam o DSDM são: envolvimento ativo do usuário, equipes com poder de decisões, entregas frequentes, aptidão para negócios, desenvolvimento incremental e iterativo, alterações reversíveis, fixar os requisitos essenciais, teste em todo o ciclo de vida e colaboração dos *stakeholders* [32].

Em síntese, os métodos de desenvolvimento ágil se esforçam para atender a imprevisibilidade na construção dos *softwares*, de modo a responder às mudanças buscando integrar todas as etapas do projeto, desde a etapa de requisitos à etapa de *design*, de implementação e de processos de teste [14].

2.2 Engenharia de Requisitos Ágil

Na Engenharia de *Software*, a atividade de Engenharia de Requisitos (*Requirements Engineering* - RE) visa apoiar o desenvolvimento de produtos que atendam às expectativas dos clientes em termos de funcionalidade e qualidade. Para isso, diversas técnicas, práticas e artefatos são indicados [64; 69]. No entanto, nem todas as abordagens adotadas no processo de RE se adaptam aos métodos de desenvolvimento de *software* existentes, como os métodos tradicionais e os métodos ágeis [1].

Em métodos ágeis, por exemplo, o foco do processo está no *software* em funcionamento e não nos artefatos gerados [42]. Embora este seja um dos objetivos mais relevantes do método ágil, é também um dos grandes desafios enfrentados pelos profissionais de requisitos,

pois gerar e manter a documentação de requisitos é visto, neste contexto, como uma prática burocrática, o que torna o processo menos ágil [56].

Nesse contexto, a Engenharia de Requisitos Ágeis (*Agile Requirements Engineering* ou *Agile RE*) tem sido um termo usado para definir a “maneira ágil” de planejar, executar e testar os requisitos especificados durante um processo ágil [44]. E, tem sido objeto de estudo de diversos pesquisadores com o intuito de compreender as técnicas e práticas de elicitação de requisitos ágeis adotadas, as limitações, desafios e implicações dessas técnicas, os métodos ágeis indicados, as formas de gerenciamento de requisitos, entre outros [44; 56; 68].

Na *Agile RE*, o projeto inicia apenas com uma visão geral do problema, sem maiores detalhes [58]. A compreensão dos requisitos ocorre ao longo do projeto, de forma iterativa e incremental [56; 58], onde são elicitados, analisados e especificados em colaboração estreita e contínua com um cliente ou representante do cliente, para obter alta reatividade a mudanças nos requisitos e, conseqüentemente, no sistema [38].

No entanto, na *Agile RE*, os requisitos permanecem sem documentação por um longo período de tempo, o que pode ser prejudicial para o projeto [3; 52]. Alsaqaf, Daneva e Wieringa [3] acrescentam que os profissionais ainda não compartilham uma perspectiva comum sobre a documentação de requisitos, embora diferentes abordagens tenham sido usadas para a Especificação de Requisitos de *Software* (*Software Requirements Specifications* - SRS), por exemplo: *User Stories*, *Use Case*, regras, protótipos e modelos [58].

Em geral, na *Agile RE*, os requisitos são representados em *User Stories*, especificamente formulados para capturar como um *stakeholder* pretende usar o sistema em desenvolvimento [37]. Nesse sentido, detalhes mais específicos dos requisitos são esclarecidos por meio de conversas entre desenvolvedores e clientes finais. É válido ressaltar que os desenvolvedores atuam como parte do papel do analista, ou seja, eles comunicam e discutem *User Stories* diretamente com os clientes finais [45].

Alguns estudos empíricos realizados na indústria perceberam que: as equipes ágeis se concentram na documentação textual dos requisitos elicitados com uma variedade de técnicas [75; 76]; testes e requisitos de *software* são atividades alinhadas [76]; requisitos obscuros e falhas na comunicação são problemas em constante evidência [13; 43]; bem como, garantia de qualidade, obtenção de requisitos de qualidade, desafios conceituais e arquitetura de *software* [3]; entrevistas, reuniões e prototipagem são as técnicas de elicitação

comumente usadas [75]; embora a prototipagem ajude em alguns dos desafios do *Agile RE*, algumas práticas complementares são necessárias, como seguir os requisitos de qualidade e usar o Desenvolvimento Orientado a Testes de Aceitação (ATDD) [52]; melhorar a qualidade SRS em Desenvolvimento Ágil de *Software* (ASD) pode ajudar a obter vantagem na competitividade da indústria de *software* [58]; entre outros resultados [45; 57; 60; 66].

Outros estudos de Revisões Sistemáticas de Literatura (RSL) apontam diversos problemas e/ou desafios relacionados ao *Agile RE*, como: o baixo envolvimento dos usuários e as constantes mudanças nos requisitos [56; 68]; o mesmo entendimento sobre o projeto, artefatos, documentação e requisitos não funcionais [68]; o gerenciamento de mudanças, ferramentas de requisitos de *software*, entre outros [29]; a priorização de requisitos, requisitos especificados informalmente, entre outros [38]; lidar com requisitos não-funcionais e equipes auto-organizadas [44]; entre outros resultados. Alguns destes estudos também destacam práticas ou propostas de solução para os problemas identificados [38; 44; 56].

2.3 Teste de *Software* no desenvolvimento de *software* ágil

O Teste de *Software* pode ser definido como uma atividade que objetiva verificar se o *software* produzido foi implementado corretamente, ou seja, se está de acordo com os requisitos especificados pelo cliente e se satisfaz suas expectativas [12]. No entanto, devido a algumas características próprias do *software*, como flexibilidade para mudanças e complexidade, o teste nem sempre é uma tarefa trivial.

Crespo et al. [25] aponta algumas dificuldades em testar *software* e destaca, dentre elas, a preocupação com a atividade de teste somente na fase final do projeto - uma característica comum em atividades de teste que são realizadas em métodos tradicionais.

Para que os testes em projetos ágeis sejam bem-sucedidos, é necessário definir e adotar estratégias de testes que abranjam todas as atividades de desenvolvimento [19] e garantam a qualidade do produto em sua entrega final.

Assim, Teste de *Software* Ágil (*Agile Software Testing* ou *Agile ST*) tem sido um termo usado para definir um conjunto de práticas que incorporam técnicas de teste, comumente

usadas, e que consideram valores ágeis para isso [26]. Em *Agile ST*, a equipe não está apenas envolvida na identificação de falhas¹, mas também na prevenção das mesmas [28; 54]. Dessa forma, *Agile ST* é um desafio para equipes que costumam realizar o teste, apenas, na entrega final do projeto; e que, em alguns casos, concentram a atividade de teste em um único profissional - o testador. No *Agile ST*, todos os membros da equipe são responsáveis pelos testes, o que impulsiona a equipe a ser proativa e começar a testar desde o início do projeto [19; 23].

Para Cruzes et al. [28], o *Agile ST* também aumenta a necessidade de melhorar a comunicação e coordenação entre os membros da equipe, bem como uma nova mentalidade pessoal e organizacional. A comunicação e a interação eficientes entre a equipe melhoram o teste e o desenvolvimento, eventualmente melhorando o fluxo e a eficiência das informações no processo [72]. Desse modo, manter a participação ativa dos membros da equipe no processo de desenvolvimento do *software*, também contribui com a execução dos testes durante o ciclo de vida do *software*.

Conforme descrito em Crispin e Gregory [26], o *Agile ST* não significa apenas testar em projetos ágeis, mas testar uma aplicação com um plano para aprender sobre ela e permitir que as informações do cliente guiem o teste, respeitando os valores ágeis. Para isso, Crispin e Gregory [26] apresentam os quatro Quadrantes para Testes Ágeis (*Agile Testing Quadrants*) (ver a Figura 2.1).



Figura 2.1: Quadrantes para Testes Ágeis (Adaptado de Crispin and Gregory [26]).

¹Falha: se refere a incapacidade do *software* de realizar a função requisitada [69].

Cada um dos quatro quadrantes reflete as diferentes razões pelas quais os testes devem ser realizados. Crispin e Gregory [26] explicam o significado de cada quadrante da seguinte forma:

- O quadrante inferior esquerdo (Quadrante 1 ou Q1) representa o Desenvolvimento Orientado a Testes, que é uma prática central do desenvolvimento ágil. Os testes neste quadrante, teste de unidade e teste de componente, são chamados de teste de programador ou teste de tecnologia.
- O quadrante superior esquerdo (Quadrante 2 ou Q2) refere-se aos testes que suportam o trabalho da equipe de desenvolvimento, mas em alto nível. Esses testes orientados aos negócios, também chamados de testes voltados para o cliente e testes do cliente, definem a qualidade externa e os recursos que os clientes desejam.
- O quadrante superior direito (Quadrante 3 ou Q3) consiste em testes que investigam se o produto atende as necessidades do usuário. Nesse quadrante o objetivo é criticar o produto implementado, gerando informações que guiem o desenvolvimento. Assim, encontrar falhas ou cenários omitidos é uma das características principais deste quadrante.
- O quadrante inferior direito (Quadrante 4 ou Q4) trata de testes focados em tecnologia e visa criticar características do produto como desempenho, robustez e segurança, por meio da análise do comportamento do sistema em ambiente de produção.

Alguns estudos empíricos investigaram as atividades do *Agile ST* em um contexto de indústria e descobriram: cinco principais desafios enfrentados por profissionais ágeis e diferentes estratégias para superar esses desafios enquanto praticam a automação de testes [72]; que existe a necessidade de um apoio mais específico para o planejamento e execução do Testes Exploratórios [23]; que ET é útil na detecção de erros² distintos, não encontrados durante testes baseados em casos de teste [48]; uma lista de nove fatores-chave, agrupados em quatro temas, que permitem a eficiência e eficácia de ET em sistemas de *software* em larga escala [55]; dentre outros [28; 30]. Um outro estudo realizou uma comparação entre os mé-

²Erro: é caracterizado como uma ação executada no *software*, que produz um resultado incorreto [26].

todos ágeis *XP*, *SCRUM* e *Kanban*, e concluiu que embora esses métodos sejam diferentes em alguns papéis e práticas, nas atividades de teste são semelhantes [8].

Outros estudos de Revisão Sistemática da Literatura (RSL) mostraram a necessidade de ferramentas de teste no contexto de métodos ágeis [17; 33]; os principais fatores técnicos, culturais e gerenciais que impulsionam os testes em ambientes *DevOps* [4]; fatores que limitam a adoção do Desenvolvimento Orientado a Testes (TDD) pelo setor [17]; a necessidade de técnicas de gerenciamento de Teste Exploratório mais voltada às necessidades das equipes [33].

2.4 Testes Exploratórios

O Teste Exploratório (*Exploratory Testing* - ET) ganhou impulso nas últimas décadas, por se caracterizar como uma forma simples de explorar um sistema enquanto se aprende sobre ele [23]. Essa compreensão do sistema é derivada de várias fontes: observação do comportamento do produto durante o teste, familiaridade com o aplicativo, a plataforma, o processo de falha, o tipo de possíveis falhas, entre outros [2].

O termo Teste Exploratório foi introduzido por Kaner [47] para se referir a uma abordagem de testes utilizada para testar *software* sem casos de teste pré-projetados. Com o passar dos anos, o ET evoluiu para uma abordagem cuidadosa de testes manuais [2], onde o testador planeja, testa, executa e analisa o teste, enquanto vai aprendendo e explorando o produto [6].

Atualmente, o ET é visto na indústria como uma abordagem pela qual diferentes técnicas de teste podem ser aplicadas [33; 81] e oferece grande contribuição quando o projeto não contém documentação especificando o sistema ou esta é de baixa qualidade ou desatualizada [10; 62]; ou ainda, quando o código passa por mudanças rápidas.

Uma das características que diferem ET de outros tipos de testes é a ausência de *scripts* e planos de testes bem detalhados. Em teoria, o profissional que realiza o teste não têm casos de teste predefinidos; em vez disso, ele usa suas habilidades e imaginação para encontrar defeitos³ e áreas fracas do *software* testado [65].

Para isso, algumas ações são sugeridas, como conhecer o produto, aprender os pontos

³Defeito: é a manifestação de um erro no *software*, também conhecido como *bug* e se executado, pode acarretar uma falha [69].

fracos do produto, aprender o caminho em que o produto falha, aprender a testar o produto, reportar os problemas ou fazer novos testes com o que foi aprendido até o momento [77].

Em síntese, ET é útil quando (i) é necessário obter um *feedback* rápido, (ii) os requisitos são vagos, (iii) o desenvolvimento está em uma fase inicial, (iv) o sistema é instável ou (v) um defeito é detectado e há a necessidade de explorar as variações e o escopo desse defeito [65], por exemplo.

Entretanto, como em ET o *design* e execução de testes são realizados ao mesmo tempo [77], algumas desvantagens podem ser percebidas, como: a falta de preparação, estrutura e orientação pode levar a muitas horas improdutivas [71]; uma mesma funcionalidade ser testada mais de uma vez enquanto outras não serem testadas [33], especialmente quando vários testadores ou equipes de teste estão envolvidos; dificuldade de acompanhar o progresso dos profissionais de teste [71]; dentre outras.

Portanto, para suprir a necessidade de gerenciamento e medição do ET e tratar algumas dificuldades específicas da execução deste, algumas abordagens surgiram, como: *Session Based Test Management* (SBTM), *Exploratory Testing in Pairs*, *Team Exploratory Testing Sessions* (TET), *Tours*, *Rule-Based Exploratory Testing* (R-BET) [6; 46; 65; 77].

Entretanto, para uma adequação mais específica ao contexto prático de equipes ágeis, tais abordagens apresentam algumas limitações, como: orientação de tempo mínimo e máximo para a realização das sessões - por vezes, a sessão poderia ser mais curta quando uma funcionalidade testada é mais simples, por exemplo; alguns artefatos (como o *charters* e o relatório) adotados nas sessões de ET estão obsoletos ou possuem campos subutilizados para o registro do teste; toda a equipe dedicada à execução do ET, não otimiza o tempo hábil reservado o teste; regras muito específicas para a realização do ET, limitam a liberdade do profissional que executa o teste, na exploração do sistema; dentre outros aspectos.

2.4.1 *Session Based Test Management* (SBTM)

Como forma de suprir a necessidade de gerenciamento e medição do ET, Bach [6] propôs (1) dividir as atividades de testes em sessões, que seriam a unidade básica de trabalho, (2) estipular uma missão para cada sessão e (3) adotar métricas do tempo relacionado às atividades de teste, originando a estratégia do *Session Based Test Management* (SBTM).

As principais características do SBTM [6] são a definição da missão, que consiste em

descrever brevemente o objetivo do teste; o registro do tempo de duração da sessão, que deve ser em média 90 minutos, mas pode variar entre 30 e 120 minutos, de acordo com a missão; não há uma sequência de passos definidos para executar o teste; e, o registro das tarefas realizadas no ET, tais como, dados usados, *feedback* do sistema, atividades feitas, dentre outras.

É válido ressaltar que o tempo da sessão de ET, no SBTM, é definida pelo testador, de acordo com suas habilidades. Durante esse período de tempo, o testador deve concentrar-se exclusivamente na execução dos testes. Orienta-se que uma sessão de ET deve ser executada sem interrupções, durante toda a sua duração.

Ghazi [36] destaca que uma sessão de ET deve começar com um documento (o *charter*) que contém a missão descrita de maneira sucinta. O objetivo é garantir que o testador se mantenha concentrado unicamente na execução da sessão descrita no *charter*.

Algumas orientações são indicadas para definir a missão, no *charter*: (i) a missão não deve ser muito específica, nem muito genérica; (ii) a missão determina o que deve ser testado (não como o teste deve ser realizado); (iii) ao final da sessão de ET, novas ideias, oportunidades ou problemas, encontrados pelo testador, podem ser usados para a criação de novas missões; (iv) após a conclusão da missão, é importante haver uma avaliação da sessão, a fim de discutir os resultados encontrados.

Para Hendrickson [40] o formato da missão deve ser baseada na seguinte premissa: defina a missão e o que deve ser explorado. A missão de um ET, pode ser definida com a estimativa de *test points*, ou seja, pontos de teste. Um *test point* está relacionado a cada trabalho de teste executado na missão do ET. Cada missão pode conter um ou vários *test points* que deverão ser investigados durante o tempo da sessão do ET. É importante destacar que a lista de *test points* é dinâmica, ou seja, novos pontos podem ser adicionados, com base em erros encontrados e correções [36]; e, devem ser testados conforme o risco (alto, médio ou baixo), sendo os de maior risco primeiro.

De um modo geral, a estratégia do SBTM é comumente usada com a finalidade de tornar o ET com objetivos mais definidos, e de apoiar os problemas de auto-gerência e gerência da equipe. Alguns dos seus elementos, como *charters* ou a definição de sessão, são usados como fatores básicos para aplicação de ET por outras abordagens, como o *Exploratory Testing in Pairs*, o TET e o R-BET (descritas a seguir).

2.4.2 *Exploratory Testing in Pairs*

O *Exploratory Testing in Pairs* consiste em uma abordagem que exercita a prática de ET realizada por dois profissionais simultaneamente, ou seja, duas pessoas trabalham juntas para avaliar e completar um único teste [46]. Por meio do teste em par, algumas dificuldades comuns em ET são mitigadas como, pouca experiência do testador, falta de foco da equipe, desconhecimento do domínio da aplicação e reprodutibilidade dos testes.

Nesta abordagem, observam-se algumas características semelhantes ao SBTM, como o uso de *charter* de teste, a definição da missão e uma sessão que possui tempo de duração limitado. O *charter* pode conter a descrição das táticas a serem adotadas no teste, os riscos, as ferramentas serem adotadas, as saídas esperadas, dentre outras informações necessárias à execução da sessão [46].

Assim, um dos profissionais executa o teste e o outro sugere ideias ou teste, presta atenção e toma notas, ouve, faz perguntas, pega material de referência, etc. [46]. É indicado ainda que, troquem ideias sobre o teste, discutam os cenários a serem avaliados, façam anotações do que foi observado no teste e colaborem entre si durante a aplicação do ET. Os papéis podem ser revezados.

Para Kaner e Bach [46], o *Exploratory Testing in Pairs* consiste em uma “atividade de geração de ideias”, em vez de uma atividade de implementação de um plano. Por este motivo, sua implementação traz alguns benefícios, como: (i) aumenta o foco na atividade realizada; (ii) incita a criação de novas estratégias de teste; (iii) reduz interrupções enquanto o teste é executado; (iv) estimula a criatividade dos profissionais; (v) aumenta a qualidade do relatório de *bugs*, já que é revisado pelos dois profissionais; (vi) contribui no treinamento de profissionais iniciantes, por meio da troca de ideias com profissionais mais experientes; dentre outros aspectos [46].

2.4.3 **Team Exploratory Testing Sessions (TET)**

A abordagem *Team Exploratory Testing Sessions* (TET), proposta por Saukkoriipi e Tervonen [67], foi definida com base na abordagem SBTM. O principal objetivo do TET é a realização de sessões de teste com uma equipe multidisciplinar, com *skills* complementares. Em síntese, o TET considera como prioridade o objetivo do teste e sua

importância, o senso de responsabilidade pelos resultados e a autonomia sobre como atuar durante a sessão. Para isso, as sessões de ET são realizadas por uma equipe de profissionais com conhecimentos distintos [65].

Resumidamente, o processo da sessão TET consiste em três fases: preparação, sessão e finalização. Saukkoriipi e Tervonen [67] detalham a fase de preparação, em: metas, participantes, *software*, área de foco, tempo, ferramentas, técnicas de teste, ambiente de teste, treinamento, comunicação; (2) a fase de sessão, em: apresentação e suporte, a execução do ET, dar e receber *feedback*, aprendizado sobre o teste, socialização e relatório dos resultados; e, (3) a fase de finalização, em: comunicação dos resultados do teste (ideias de aprimoramento, defeitos, criação/atualização de casos de teste, reconhecimento da área de teste), relatório da sessão, *feedback*, atualização da abordagem de teste e acompanhamento.

Para aplicar o TET é indicado que os membros da equipe possuam habilidades, experiências e papéis distintos, e, de modo complementar, tenham experiência em testes e conhecimento sobre o domínio do sistema a ser testado [67]. Para isso, são indicados os seguintes papéis: facilitador, especialista do domínio, participante e gerente do domínio. A execução da sessão do ET é semelhante à sessão do SBTM, com a definição de uma missão, de um tempo e o registro do teste em um relatório.

O TET contribui com a mitigação de algumas fraquezas do ET, como a pouca experiência do testador e a falta de foco da equipe, por meio da ênfase na realização de sessões em equipe. Desse modo, a equipe permite um ganho sinérgico quando os membros interagem de maneira a ajudá-los mutuamente, aprendendo uns com os outros [67].

2.4.4 Tours

A primeira definição para *Tour Testing* foi dado por Kaner [47], definindo *Tour* como uma abordagem estruturada para ET, onde a exploração de um produto é organizada em torno de um tema específico, e em seguida, o testador pode navegar e percorrer o sistema para registrar o comportamento da funcionalidade testada [77].

O conceito de *Tours* se baseia na metáfora do turista. Enquanto o testador realiza o ET, ele se assemelha a um turista que explora um lugar onde nunca esteve antes [77]. Considerando que um turista gostaria de visitar o maior número possível de lugares, em um prazo delimitado, as *Tours* são agrupadas em seis grupos principais de acordo com suas

características [77], a citar: *Business district*, *Historical district*, *Tourist district*, *Entertainment district*, *Hotel district* e *Seedy district*.

Assim, o testador é direcionado a explorar o sistema sob a perspectiva de: funcionalidades referentes ao contexto de negócio; funcionalidade que possuem código legado, histórico de *bugs* e funcionalidades defeituosas; funcionalidades pouco utilizadas por usuários mais experientes, pois podem ser utilizadas por novos usuários; funcionalidades que suportam e complementam as outras funcionalidades; funcionalidades que estão em funcionamento mesmo com o sistema “em descanso”; encontrar vulnerabilidades no sistema [77].

Nessa perspectiva, a *Tours* contribui com a mitigação de algumas fraquezas do ET, como a pouca experiência do testador, a falta de foco e a aplicação de Testes Exploratórios a Sistemas de Tempo Real.

2.4.5 Rule-Based Exploratory Testing

A abordagem *Rule-Based Exploratory Testing* (R-BET) foi proposta por Hellman e Maurer [39] com objetivo de fornecer uma estratégia para execução de ET baseada no uso de regras curtas e automatizadas para aumentar a capacidade de detecção de *bugs* de sessões de ET gravadas [39].

Nesta abordagem, as regras seguem o formato de uma declaração “*if..try..catch*”. De acordo com Hellman e Maurer [39], a pré-condição “*if*”, indica que o teste só será executado se ela for satisfeita - neste caso, pode haver mais de uma pré-condição associadas por um “*and*”. O termo “*try*” indica uma ação e representa o corpo principal da regra que será executada através das pré-condições. O termo “*catch*” indica o que deve ocorrer se o *try* falhar (identificação de *bugs* ou *warnings*) ou lançar uma exceção.

O R-BET associa ET e Testes Baseados em Regras (*Rule-Based Testing*) para testes em interfaces de usuário (*Graphical User Interface* - GUI). Para isso, adota o uso de uma ferramenta denominada LETT (*LEET Enhances Exploratory Testing*), que captura a interação do testador durante uma sessão de ET, gerando um *script* reproduzível e, em seguida, recebe um conjunto de regras definidas pelo testador, indicando comportamentos esperados ou proibidos para a funcionalidade testada [39].

Para Hellman e Maurer [39], o R-BET permite que um sistema seja testado mais detalhadamente e propicia que os testadores realizem testes mais interessantes. Neste sen-

tido, o R-BET contribui com a mitigação de uma das desvantagens do ET que se refere à reprodutibilidade dos testes.

2.5 Design Science Research

A *Design Science Research* (DSR) consiste em um método de pesquisa orientado à resolução de problemas. Para Wieringa [78], a DSR envolve a pesquisa na resolução de situações-problema em que ciências tradicionais não sejam suficientes para sustentar o processo de pesquisa.

Dresch et al. [31] complementa que, as características apresentadas pela DSR indicam a possibilidade de utilizar os fundamentos desta ciência para a criação de conhecimento aplicável às organizações. Desse modo, a DSR difere de outras metodologias de pesquisa porque ela é orientada à resolução de problemas específicos para obter uma solução satisfatória para a situação, mesmo que a solução não seja ótima [31].

Dois fatores são fundamentais para o desenvolvimento da pesquisa [41]: a relevância, que orienta o desenvolvimento de soluções baseadas em tecnologia para problemas gerenciais importantes e relevantes; e, o rigor, que sugere que a solução passe por métodos rigorosos, tanto na construção como na avaliação dos artefatos.

Para Hevner [41], o objetivo de aplicar DSR é desenvolver artefatos que permitam soluções satisfatórias aos problemas práticos. Desse modo, a busca por um artefato eficaz e efetivo exige a utilização de meios que sejam disponíveis, para alcançar fins desejados [41], ao mesmo tempo que deve satisfazer o contexto do ambiente em que o problema está sendo estudado.

Portanto, com base em um rigor metodológico, a DSR orienta as etapas de um projeto como uma atividade científica, e proporciona a elaboração e a avaliação minuciosa de um artefato viável, que pode ser definido como um *constructo*, modelo, método e/ou instanciação [78].

Em síntese, Cross [27] afirma que: um *Constructo* consiste em artefatos que definem conceitos para descrever os problemas dentro de um domínio e para especificar as possíveis soluções - em linhas gerais, formam o vocabulário de um domínio; um *Modelo* se refere à um conjunto de proposições ou declarações que expressam as relações entre os *constructos*;

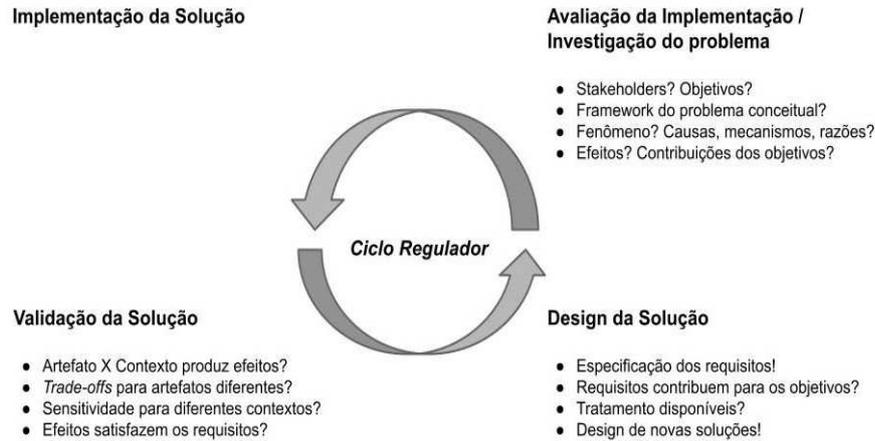


Figura 2.2: Ciclo Regulador da DSR, proposto por Wieringa [78].

um *Método* consiste em um conjunto de passos (um algoritmo ou orientação) usado para executar uma tarefa; e, as *Instanciações* operacionalizam os *constructos*, os modelos e os métodos.

Desse modo, a utilidade, qualidade e eficácia do artefato devem ser, rigorosamente, demonstradas por meio de métodos de avaliação bem executados [31]. Mesmo gerando resultados a partir de um processo, o método contribui para fortalecer a base de conhecimentos existentes, podendo colaborar para o aprimoramento de teorias [41].

Para isso, Wieringa [78] categoriza a DSR em etapas que encadeiam um “Ciclo Regulador”. Este ciclo pode ser repetido quantas vezes forem necessárias até o refinamento completo do artefato, ou seja, até que o problema encontrado na etapa de Investigação seja solucionado.

Em síntese, o “Ciclo Regulador”, proposto por Wieringa [78], é baseado em um conjunto de atividades científicas que norteiam o planejamento, desenvolvimento e avaliação do artefato, por meio das seguintes etapas: Investigação do Problema, *Design* da Solução, Validação da Solução, Implementação da Solução e Avaliação da Implementação, conforme apresentado na Figura 2.2.

O Ciclo Regulador inicia com a *Investigação de um Problema* real, visando descobrir quais fenômenos devem ser melhorados e porque devem ser melhorados. Seguido do *Design da Solução*, que consiste no *design* de um ou mais artefatos que podem solucionar o problema identificado na etapa anterior; e da *Validação da Solução*, que visa autenticar se a solução contribui com os objetivos dos *stakeholders*. Em seguida, a *Implementação da So-*

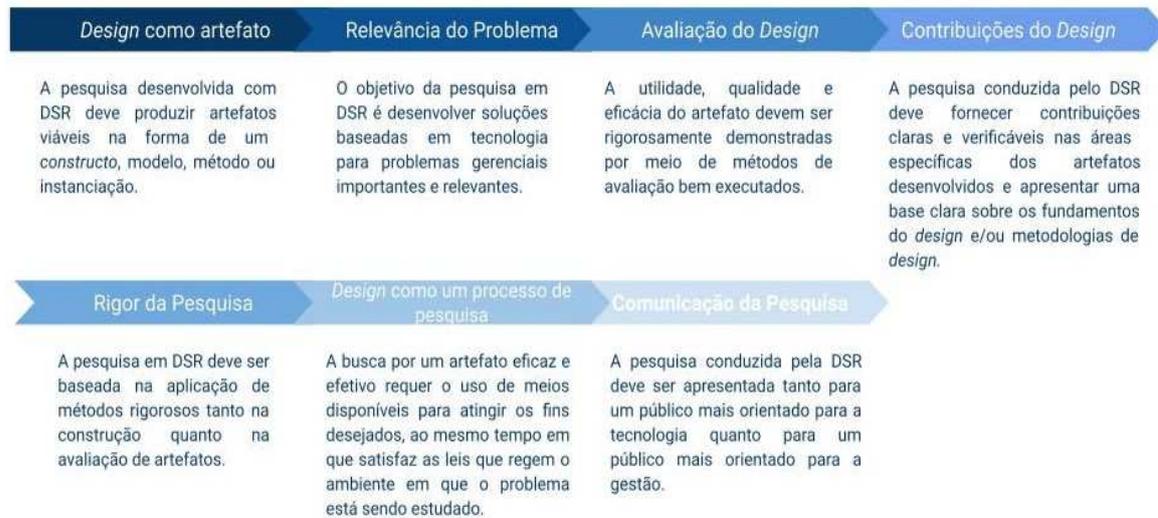


Figura 2.3: Instruções gerais para a condução da *Design Science Research* (Traduzido de Dresch, Lacerda e Jr. [31]).

lução propõe a aplicação da solução em um contexto real para que possa ser avaliada, dando início à etapa de *Avaliação da Implementação* para perceber o sucesso da solução [78].

Embora as etapas de Investigação do Problema e Avaliação da Implementação tenham as mesmas características, no que se refere às questões norteadoras, possuem objetivos distintos - como já descrito anteriormente. A Investigação do Problema integra a primeira iteração do Ciclo Regulador e consiste em investigar na literatura o problema a ser tratado na pesquisa, de modo a preparar o *design* da solução [78]. Enquanto que a Avaliação da Implementação finaliza a primeira iteração do ciclo e pode originar uma nova iteração do Ciclo Regulador. Seu objetivo consiste em avaliar a aplicação da solução em um contexto do problema investigado [78]. No Capítulo 4 desta tese, cada etapa da DSR conduzida nesta pesquisa é instanciada e detalhada.

Portanto, uma pesquisa fundamentada em DSR deve prover contribuições claras e verificáveis nas áreas específicas dos artefatos desenvolvidos, e apresentar um embasamento claro em fundamentos ou metodologias de *design* [41]. A Figura 2.3 apresenta algumas instruções gerais fornecidas por Hevner [41] e adaptadas por Dresch, Lacerda e Jr. [31].

2.6 Considerações Finais do Capítulo

Neste capítulo foram apresentadas as bases teóricas necessárias para a compreensão da pesquisa realizada nesta tese, a fim de propiciar ao leitor os conhecimentos fundamentais para interpretação dos resultados alcançados. A Seção 2.1 apresentou os conceitos e os princípios do Método Ágil na Engenharia de *Software*, bem como a descrição de algumas principais metodologias que visam apoiar o desenvolvimento de *software*.

A Seção 2.2 discutiu as principais definições sobre *Agile RE*, bem como as estratégias de especificação e documentação mais adotadas neste contexto, destacando estudos empíricos e de RSLs já realizadas.

A Seção 2.3 apresentou os principais conceitos e práticas de *Agile ST* que mais contribuem para o desenvolvimento de *software* por profissionais ágeis. Também foram destacados estudos empíricos e de RSLs realizados nesta área de estudo. De modo complementar, foi enfatizada a relevância dos ET no ambiente ágil, destacando algumas principais abordagens de ET.

A Seção 2.4 expõe uma contextualização sobre Testes Exploratórios e discute algumas principais abordagens que apoiam o gerenciamento e medição do ET, como *Session Based Test Management (SBTM)*, *Exploratory Testing in Pairs*, *Team Exploratory Testing Sessions (TET)*, *Tours* e *Rule-Based Exploratory Testing (R-BET)*.

A Seção 2.5 define brevemente os principais conceitos relacionados a *Design Science Research*. Este é o procedimento metodológico adotado nesta pesquisa e é detalhado com mais precisão no Capítulo 4.

Por fim, o próximo capítulo apresentará o artefato gerado nesta tese, o método *Agile ETeasy*. Desse modo, serão apresentados e discutidos os elementos e artefatos que compõem o referido método.

Capítulo 3

O Método *Agile ETeasy*

Neste capítulo é apresentado o artefato gerado nesta tese, de acordo com a proposta de *Design Science Research* [78], seguida neste estudo. Portanto, o artefato é definido como um método e sua construção teve início na etapa de *Design da Solução* do Ciclo 1 da DSR (conforme apresentado na Seção 4.2.2). Após os sucessivos refinamentos aplicados no artefato, durante a DSR, foi gerada a versão final do método *Agile ETeasy*. Assim sendo, os elementos que compõem as etapas de Planejamento, Execução e Encerramento do método são detalhados, respectivamente, nas Seções 3.1, 3.2 e 3.3. Posteriormente, algumas orientações adicionais para aplicação do método são fornecidas, na Seção 3.4. E, por fim, as considerações finais deste capítulo são apresentadas na Seção 3.5.

3.1 Método *Agile ETeasy*: Etapa de Planejamento

O início do método *Agile ETeasy* ocorre pela etapa de **Planejamento**, cujo objetivo principal é um plano simplificado com o escopo do teste a ser feito e que guiará a execução do método. Participa desta etapa o principal responsável pelo projeto, ou seja, o líder da equipe ou um membro da equipe com conhecimento sobre o escopo do projeto. A Figura 3.1 apresenta a atividade, as tarefas, as técnicas e o elemento de saída sugeridos para esta etapa de Planejamento.

Na etapa de Planejamento do método *Agile ETeasy* foi apontado apenas uma atividade: “Definir Escopo de Teste”. Para o desenvolvimento desta atividade serão realizadas duas tarefas, conforme apresentado na Tabela 3.1.

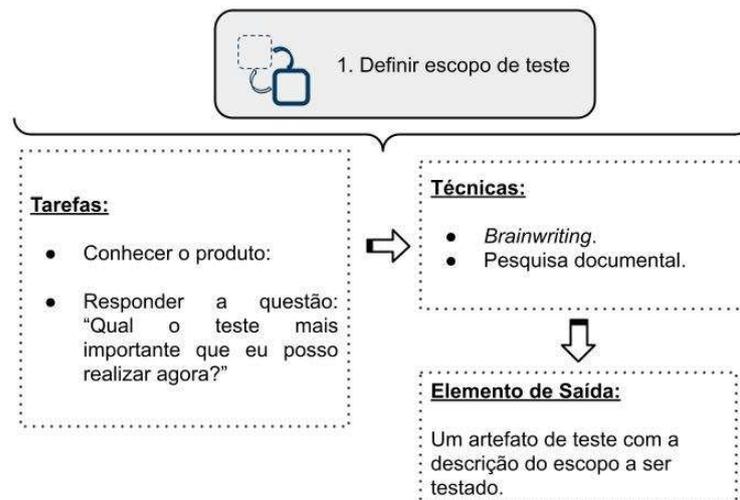


Figura 3.1: Etapa de Planejamento do método *Agile ETeasy*.

Para executar a atividade “Definir Escopo de Teste” é indicado o uso de técnicas, a citar: (i) Pesquisa Documental, por meio da descrição dos requisitos em um Sistema de Gerenciamento de Requisitos ou em um documento/modelo de requisitos específico (*Use Case*, *User Stories*, Protótipos, etc.), Manual do Usuário, dentre outros documentos disponíveis sobre o projeto e que possam úteis para auxiliar na definição do escopo de teste; e, (ii) *Brainwriting*, que consiste em uma técnica para geração de ideias, por meio da escrita de uma ideia ou opinião sobre uma questão específica, por alguns minutos, sem falar. Neste sentido, a Pesquisa Documental contribui para que o profissional conheça, rapidamente, um pouco sobre o produto que irá testar; enquanto que, o *Brainwriting* contribui com a descrição do escopo do que será testado durante a aplicação do teste.

Ao final da etapa de Planejamento, deve ser gerado, como artefato inicial, um plano¹ com o escopo do que será testado durante a execução do método *Agile ETeasy*. Para isso, indica-se que seja adotado um artefato de planejamento para cada funcionalidade testada com o método. Neste artefato, algumas informações básicas sobre o projeto devem ser inseridas (consultar o Apêndice K):

- *Data*: Consiste em informar a data em que o teste foi realizado - no formato <dd/mm/aaaa>.

¹Template do artefato de Planejamento do método *Agile ETeasy*: <https://sites.google.com/view/metodoagileeetasy/artefatos?authuser=3>

Tabela 3.1: Atividade 1. Definir escopo de teste.

| Tarefa | Descrição |
|---|--|
| Conhecer o produto | É interessante que o profissional que fará o teste reúna informações sobre a aplicação; conheça problemas passados ou conhecidos, o que aconteceu em testes anteriores, mudanças recentes, <i>etc.</i> ; e, participe de reuniões com os <i>stakeholders</i> (líder, cliente, <i>etc.</i>). |
| Responder a questão: “ <i>Qual o teste mais importante que eu posso realizar agora?</i> ” | Diante das informações coletadas na tarefa anterior, é preciso priorizar o teste a ser realizado, conforme a relevância da funcionalidade para projeto. Para isso, é indicado informar o objetivo do teste, como escopo do que será testado no método. Caso o profissional considere a ‘profundidade’ do teste muito extensa, sugere-se detalhar a descrição do escopo em objetivos menores, para facilitar a execução dos testes, posteriormente, e para que o foco seja mantido. |

- *Sistema*: Consiste em informar a identificação do nome do sistema/projeto a ser testado.
- *Versão*: Consiste em informar a versão do sistema/projeto a ser testado.
- *Sprint*: Consiste em caracterizar a *Sprint*, por meio de um nome, numeração ou outra referência.
- *Testadores*: Consiste na identificação do profissional (is) responsável (is) por testar o referido escopo.
- *ID*: Consiste em informar um identificador ao cenário de teste especificado.
- *Referência ao Requisito*: Consiste em descrever uma referência ao requisito testado - pode ser um identificador de *Use Case* ou de *User Stories*, a um documento, ou a um Sistema de Gerenciamento de Requisitos usado pela equipe.

- *Cenário de Teste*: Consiste em descrever o escopo a ser testado, em uma frase curta.

3.2 Método Agile ETeasy: Etapa de Execução

Após o planejamento do escopo de teste, a segunda etapa do método *Agile ETeasy* é a **Execução** do Teste. Esta etapa foi categorizada em duas atividades: “Aplicar os Testes” e “Observar os Resultados”. Participam desta etapa os profissionais envolvidos no desenvolvimento da aplicação e que executarão os testes - um desenvolvedor e o líder da equipe. A Figura 3.2 apresenta a atividade, as tarefas, as técnicas e o elemento de saída sugeridos para esta etapa de Execução.

Para “Aplicar os Testes” é sugerida a realização de uma única tarefa, conforme apresentado na Tabela 3.2. Nesta atividade, é possível consultar uma Lista de Heurísticas [40] com sugestões de dados a serem testados, de acordo com cada escopo definido no Planejamento. Além disso, a partir desta lista, também é possível aprender e identificar possíveis novos testes a serem executados. Esta atividade não gera nenhum elemento de saída, pois é dedicada exclusivamente à exploração do sistema.

Em seguida, deve-se “Observar os Resultados”. Para esta atividade, é sugerida a realização de uma única tarefa: analisar o comportamento do *software*, conforme descrito na Tabela 3.3.

Para executar a atividade “Observar os Resultados” é indicado o uso da técnica *Pair Review*. Para aplicação desta técnica os profissionais envolvidos no teste (líder do equipe e desenvolvedor), dividem-se em dois papéis: um, analisa o comportamento do *software*, em

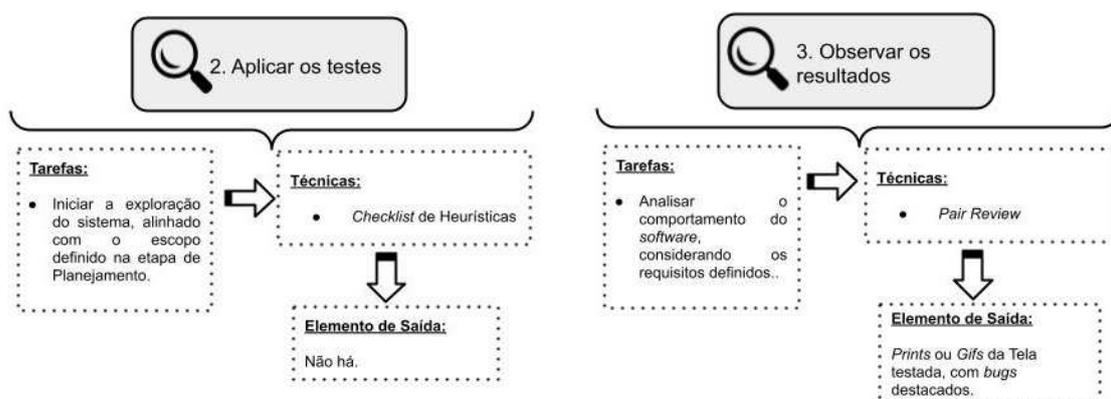


Figura 3.2: Etapa de Execução do método Agile ETeasy.

Tabela 3.2: Atividade 2. Aplicar os testes.

| Tarefa | Descrição |
|---------------------------------|--|
| Iniciar a exploração do sistema | Alinhado com o escopo definido na etapa de Planejamento, o profissional deve iniciar a exploração do sistema, considerando aspectos como pensamento crítico sobre o que está testando - enquanto também aprende sobre o sistema. |

voz alta para evidenciar seus pensamentos e impressões; enquanto que o outro, observa e fornece suas opiniões e *feedback*, simultaneamente. É sugerido também, como elemento de saída, que *prints* ou *gifs* da tela testada sejam capturados e registrados, dando destaque e reportando os defeitos encontrados.

É importante destacar que, embora as atividades “Aplicar os Testes” e “Observar os Resultados” sejam definidas separadamente na etapa de Execução, são executadas em paralelo e iterativamente.

Tabela 3.3: Atividade 3. Observar os Resultados.

| Tarefa | Descrição |
|---|--|
| Analisar o comportamento do <i>software</i> | Tendo iniciado a exploração do sistema, é indicado que os profissionais envolvidos no teste realizem a análise do comportamento (melhorias, <i>bugs</i> , novas funcionalidades, etc.) do <i>software</i> , com base nos requisitos definidos. |

3.3 Método Agile ETeasy: Etapa de Encerramento

Após a etapa de Execução, temos a etapa de **Encerramento** do método *Agile ETeasy*. Esta etapa foi categorizada em duas atividades: “Registrar o Teste” e “Alinhar Melhorias”. A Figura 3.3 apresenta a atividade, as tarefas, as técnicas e o elemento de saída sugeridos para

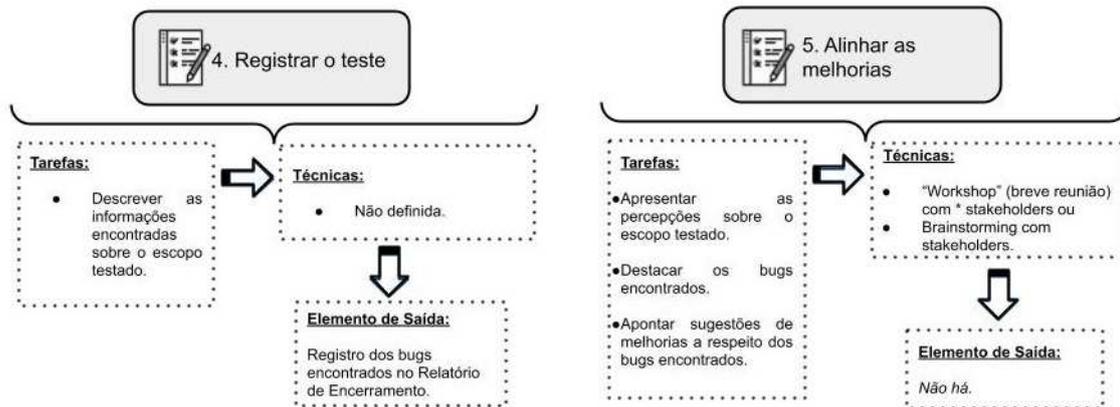


Figura 3.3: Etapa de Encerramento do método Agile ETeasy.

esta etapa de Encerramento. Desse modo, no Encerramento do teste, a atividade “Registrar o Teste” é a primeira a ser realizada. Para isso, é sugerida uma única tarefa, conforme apresentado na Tabela 3.4.

Para realizar esta tarefa não há uma sugestão de técnica a ser adotada. Portanto, é necessário que todas as observações, sobre o comportamento do sistema - durante o Teste Exploratório - sejam registradas no Relatório de Encerramento (elemento de saída do método)². Neste artefato (consultar Apêndice L), algumas informações sobre o teste devem ser inseridas, como:

- **Testadores:** Consiste na identificação do profissional (is) responsável (is) por realizar o teste.
- **Duração:** Consiste no tempo de execução (em minutos) do teste.

²Template do artefato de Encerramento do método Agile ETeasy: <https://sites.google.com/view/metodoagileeteasy/artefatos?authuser=3>

Tabela 3.4: Atividade 4. Registrar o teste.

| Tarefa | Descrição |
|---|---|
| Descrever as informações encontradas sobre o escopo testado | É indicado detalhar as observações sobre o comportamento (melhorias, <i>bugs</i> , novas funcionalidades, etc.) do sistema, de acordo com o escopo definido no teste. |

- *Escopo testado*: Consiste na descrição do escopo a ser testado - assim como está no artefato de Planejamento.
- *Usuários*: Consiste em informar o nível de acesso do usuário na funcionalidade testada.
- *Configuração*: Consiste em informar a configuração do sistema, em que o teste foi executado, se houver. Ou informações de configuração de *hardware* que são importantes de serem informadas para o entendimento do teste.
- *Arquivos*: Consiste em informar os arquivos necessários para execução da funcionalidade testada, como arquivos de carga, de entrada/saída de dados, dentre outros referentes ao contexto do sistema, e se houver.
- *Outros*: Consiste em descrever outras informações importantes a serem registradas, como: dispositivos de *hardware*/equipamentos usados no teste, dentre outros.
- *Defeitos*: Consiste em informar a descrição dos defeitos encontrados, se houver.
- *Melhorias*: Consiste em informar as sugestões de melhorias a serem implementadas a partir dos defeitos percebidos, se houver.
- *Pendências*: Consiste em informar as funcionalidades que não foram finalizadas, ou seja, as pendências percebidas durante o teste, se houver.
- *Notas*: Consiste em informar notas adicionais sobre o teste executado, se houver. Informações importantes para a equipe.
- *Anexos*: Consiste em informar os arquivos (documentos ou *links*) adicionais, necessários à execução ou compreensão da funcionalidade testada, ou de acesso à registros do teste, como *gifs* ou *prints* de tela dos *bugs* encontrados.

É importante destacar que, quando não houverem informações a serem inseridas nos campos do Relatório de Encerramento, deve-se inserir a informação “*Não foi identificado um <usuário, configuração, arquivo, defeito, melhoria, pendência, nota ou anexo>*”, no respectivo campo.

Já a atividade “Alinhar as Melhorias”, consiste em uma discussão e alinhamento com os *stakeholders* do projeto sobre as observações percebidas durante o Teste Exploratório, de modo que as decisões sobre o sistema sejam de comum acordo entre os *stakeholders*, e todos tenham o mesmo entendimento. Portanto, é necessário o desenvolvimento de algumas tarefas, conforme apresentadas na Tabela 3.5.

Tabela 3.5: Atividade 5. Alinhar as melhorias.

| Tarefa | Descrição |
|---|---|
| Apresentar as percepções sobre o escopo testado | Retornar à descrição do escopo de teste, no artefato de Planejamento, e apresentar uma síntese das observações feitas sobre a funcionalidade testada, considerando o seu comportamento durante o teste. |
| Destacar os <i>bugs</i> encontrados | Para tornar mais visível e facilitar a compreensão dos <i>bugs</i> encontrados no escopo testado é indicado expor os <i>gifs</i> ou <i>prints</i> de tela capturados na etapa de Execução, e/ou registrados no Relatório de Encerramento. |
| Apontar sugestões de melhorias a respeito dos <i>bugs</i> encontrados | Para corrigir os <i>bugs</i> encontrados, deve-se discutir o comportamento do sistema, sob a perspectiva de melhorias ou novas funcionalidades a serem implementadas. |

Para realizar a atividade “Alinhar as Melhorias” é indicado que um *Workshop* (breve reunião) ou um *Brainstorming* com os *stakeholders* do projeto seja realizado. Para conduzir esta discussão, sugere-se que as informações sobre o comportamento do sistema, na execução do escopo testado, sejam expostas aos *stakeholders*, de modo a estimular ideias sobre melhorias no sistema e/ou novas funcionalidades a serem implementadas com base nos *bugs* detectados. Nesta atividade não há indicação de um elemento de saída.

3.4 Orientações Adicionais do Método Agile ETeasy

Para bem executar o método *Agile ETeasy* algumas orientações adicionais são recomendadas, como:

1. O Teste em Par (*Pair Review*) favorece o aprendizado do testador, ao colocar profissionais com experiências diferentes, sobre o projeto, para trabalharem juntos. Bem como, contribui com a disciplina do profissional, evitando que a distração aconteça, ou seja, um ajuda o outro a manter o foco no teste. E, conseqüentemente, otimiza a qualidade do trabalho desenvolvido, garantindo a eficácia do teste executado. Neste contexto, o *Pair Review*: favorece a identificação de defeitos mais cedo/rápido (efetivo); exige pouco esforço (menor custo e pouco tempo); propicia a aplicabilidade das técnicas de apoio a execução do teste, indicadas no método *Agile ETeasy*; contribui com a identificação de mais defeitos que outros métodos de teste; auxilia na identificação de defeitos de Interface de Usuário (*Graphical User Interface - GUI*) e usabilidade, adicionalmente; encontra falhas críticas mais cedo, defeitos óbvios/simples e menos falsos positivos [33]. Assim, devem estar envolvidos no teste, o profissional que fará o Teste Exploratório (desenvolvedor que não trabalhou na implementação da funcionalidade testada) e o líder da equipe (observador).
2. Embora seja orientado que o método *Agile ETeasy* adote a técnica de *Pair Review* durante sua execução, todos os membros da equipe podem estar envolvidos. Neste sentido, indica-se que os papéis sejam distribuídos aos profissionais de acordo com as etapas do método. Na etapa de Planejamento, é indicado que o principal responsável pelo sistema (líder da equipe) planeje o escopo do teste, já que possui um conhecimento mais aprofundado sobre o sistema; enquanto que o outro profissional (desenvolvedor) observa ou opina, quando necessário, sobre o que está sendo planejado. Na etapa de Execução, é indicada a participação de dois profissionais: um desenvolvedor, que executará o Teste Exploratório; e, o líder da equipe, que observará a execução do teste e verificará se está de acordo com o escopo do teste, especificado na etapa de Planejamento. Na etapa de Encerramento, temos a participação dos profissionais que executaram o teste (desenvolvedor e líder da equipe), mais especificamente na atividade “Registrar o Teste”; e, a participação dos *stakeholders* do projeto, na atividade

- “Alinhar as Melhorias”.
3. Para apoiar a execução e o registro de Testes Exploratórios, existem algumas ferramentas disponíveis em modo *Freeware* ou *Trial*. Portanto, é sugerido o uso de ferramentas, como estas, para apoiar o desenvolvimento das etapas de Execução e/ou Encerramento do método *Agile ETeasy*. No Apêndice B é disponibilizado o acesso a uma lista de ferramentas que apoiam a execução de Testes Exploratórios. Entretanto, destacamos que: (i) nenhuma das ferramentas apresentadas executa automaticamente o método aqui proposto; (ii) e, o método *Agile ETeasy* pode ser usado independente de ferramentas específicas para Testes Exploratórios.
 4. O método *Agile ETeasy* é mais indicado para ser executado em funcionalidades que são complexas e cujos requisitos são vagos ou mudam constantemente, tornando inviável escrever casos de teste.
 5. É indicado que o método *Agile ETeasy* seja aplicado (etapa de Execução) por profissionais que não desenvolveram a funcionalidade a ser testada - definida no artefato de Planejamento.
 6. Sobre o artefato Relatório de Encerramento: é necessário gerar um relatório para cada escopo de teste definido na etapa de Planejamento.
 7. É indicado que a técnica de *Brainstorming* seja adotada na etapa de Encerramento, apenas se a funcionalidade testada necessitar de ampla discussão, a partir dos defeitos, melhorias ou pendências percebidas durante o teste. Caso, contrário indica-se a utilização da técnica de *Workshop*.
 8. É indicado que o método *Agile ETeasy* seja aplicado sempre que uma funcionalidade for implementada na *sprint*.
 9. No contexto do método *Agile ETeasy*, considera-se como “funcionalidade”: um requisito de *software* que foi implementado e caracteriza o funcionamento do sistema; e, como “escopo”: o conjunto de verificações que serão testadas acerca da funcionalidade.

10. No contexto ágil é comum que a equipe não possua um profissional dedicado exclusivamente aos testes; por este motivo, o método *Agile ETeasy* indica que os testes podem ser executados pelos desenvolvedores ou pelo líder da equipe.
11. Sempre que um defeito for corrigido na funcionalidade testada e for necessário testá-la novamente, um novo Relatório de Encerramento deve ser gerado para registrar o comportamento da funcionalidade.

3.5 Considerações finais do capítulo

Este capítulo apresentou o artefato gerado nesta tese, o método *Agile ETeasy*. Este método consiste na definição de etapas, atividades, tarefas, técnicas e elementos de saída que facilitam a operacionalização dos Testes Exploratórios por equipes ágeis.

A Seção 3.1 descreveu a atividade “Definir escopo de teste”, destacando as tarefas, as técnicas e o elemento de saída que compõem a etapa de Planejamento do método *Agile ETeasy*. Foi apresentado, também, o artefato de Planejamento indicado para esta etapa do método.

A Seção 3.2 descreveu as atividades “Aplicar os Testes” e “Observar os Resultados”, destacando as tarefas, as técnicas e o elemento de saída que compõem a etapa de Execução do método *Agile ETeasy*.

A Seção 3.3 descreveu as atividades “Registrar o Teste” e “Alinhar Melhorias”, destacando as tarefas, as técnicas e o elemento de saída que compõem a etapa de Encerramento do método *Agile ETeasy*. Foi apresentado, também, o artefato de Encerramento indicado para esta etapa do método.

A Seção 3.4 expôs algumas principais orientações para aplicação do método *Agile ETeasy*, com base nas experiências conduzidas nos estudos qualitativos realizados para validação e avaliação do método - detalhados no capítulo seguinte.

Por fim, o próximo capítulo discutirá a metodologia de *Design Science Research*, adotada para o desenvolvimento desta tese. Serão apresentados os estudos conduzidos e os resultados obtidos em cada etapa da DSR.

Capítulo 4

Aplicação da *Design Science Research*

Neste capítulo é apresentado o detalhamento das etapas da *Design Science Research* conduzida nesta pesquisa. Desse modo, são expostas algumas características gerais da metodologia da DSR aplicada ao contexto deste estudo, na Seção 4.1. Em seguida, são descritas as atividades de pesquisa realizadas em cada ciclo da DSR. Assim, as etapas que compõem o Ciclo 1 da DSR (*Investigação do Problema, Design da Solução, Validação da Solução, Implementação da Solução*) são descritas na Seção 4.2. E, as etapas que compõem o Ciclo 2 da DSR (*Avaliação da Implementação, Design da Solução, Validação da Solução, Implementação da Solução*) são discutidas na Seção 4.3. Por fim, as considerações finais deste capítulo são apresentadas na Seção 4.4.

4.1 Metodologia da DSR

O método de pesquisa baseado em *Design Science Research* visa garantir o rigor na condução da pesquisa e a relevância de seus resultados, aplicados a um contexto específico [41]. Portanto, para garantir que a pesquisa é potencialmente relevante e que é suscetível de discussão e verificação, a DSR propõe a execução de um processo sistemático e iterativo, baseado em um ou mais ciclos de tarefas. Tais tarefas consideram um conjunto de estratégias de pesquisa, planejadas e executadas, para garantir a solução de um dado problema. Por este motivo, diante do problema abordado nesta pesquisa, dos objetivos e das questões de pesquisa a serem investigadas, optou-se pelo uso deste procedimento metodológico.

Assim, para a construção e avaliação do método como artefato gerado nesta tese, foram

realizados dois ciclos da DSR [78]. A Figura 4.1 ilustra as atividades desenvolvidas em cada etapa dos ciclos, desde a Investigação do Problema à Avaliação da Implementação, as quais são descritas nas subseções seguintes.

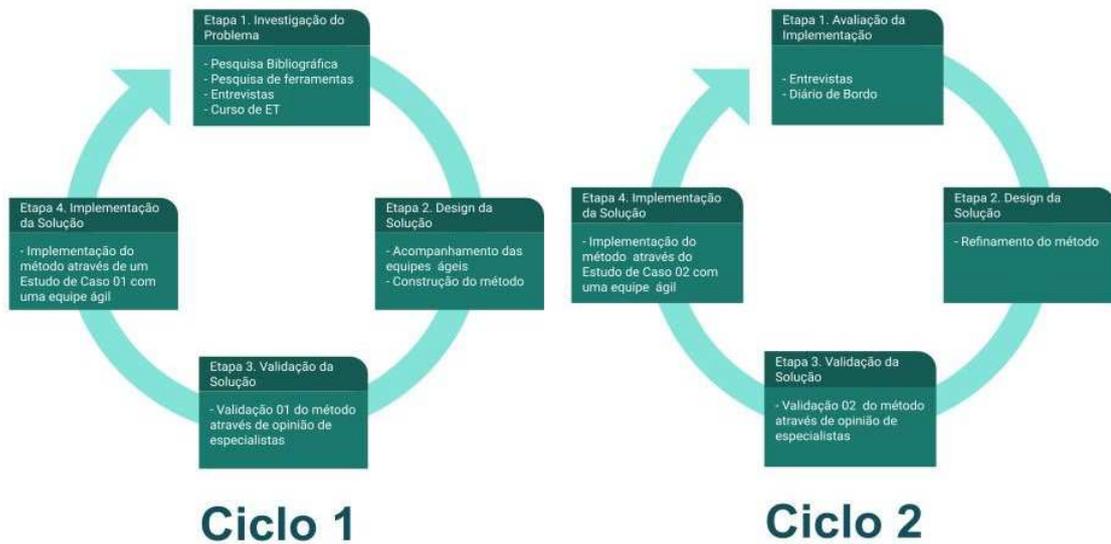


Figura 4.1: Ciclos da DSR aplicados nesta pesquisa.

O Ciclo 1 foi iniciado com a etapa de *Investigação do Problema*, que consistiu na realização de (i) uma pesquisa bibliográfica sobre ET no contexto de desenvolvimento ágil, (ii) uma pesquisa complementar por ferramentas de ET e (iii) entrevistas com profissionais de equipes ágeis, que culminou na realização de (iv) um curso sobre ET com profissionais de equipes ágeis [23]. O detalhamento da metodologia adotada nesta etapa de *Investigação do Problema* é descrito na Seção 4.1.1.

Para a realização da etapa de *Design da Solução* do Ciclo 1 foi necessário, primeiramente, (i) acompanhar a aplicação de ET em projetos reais desenvolvidos por profissionais de equipes ágeis; em seguida, coletar mais *insights* por meio de um (ii) *brainstorming*; e, finalmente, iniciar a construção do método. A metodologia utilizada na etapa de *Design da Solução* é descrita na Seção 4.1.1.

Ainda no Ciclo 1, na etapa de *Validação da Solução* foi conduzido um estudo de verificação empírica com profissionais especialistas da área, por meio de um questionário *online*. A metodologia usada na etapa de *Validação da Solução* pode ser acompanhada na Seção 4.1.1.

Em seguida, foi aplicada a etapa de *Implementação da Solução*, com o intuito de analisar a facilidade de entendimento e de uso do método proposto. Assim, foi conduzido o Estudo

de Caso 1 com uma equipe ágil, composta por três profissionais. Os procedimentos metodológicos adotados nesta etapa são descritos na Seção 4.1.1, pois compõem o refinamento do método.

Já o Ciclo 2 foi iniciado com a etapa de *Avaliação da Implementação*, cujos resultados foram obtidos a partir do Estudo de Caso 1 e, posteriormente, do Estudo de Caso 2. Para isso, foram aplicados um questionário *online* semi-estruturado para caracterização dos participantes, um diário de bordo e entrevistas com os participantes, ao final de cada Estudo de Caso. A metodologia aplicada na etapa de *Implementação da Solução* é descrita na Seção 4.1.2.

A etapa de *Design da Solução*, do Ciclo 2, foi conduzida a partir dos dados e das informações levantadas no Estudo de Caso 1, que resultaram no refinamento da solução e, conseqüentemente, em uma nova versão do método. A metodologia adotada nesta etapa de *Design da Solução* é descrita na Seção 4.1.2.

A etapa de *Validação da Solução*, do Ciclo 2, também consistiu em um estudo de verificação empírica com opinião de profissionais especialistas da área. A partir da análise da solução proposta, os profissionais foram conduzidos à expor sua opinião crítica por meio de um questionário *online* semi-estruturado. Os resultados dessa validação embasaram um novo refinamento da solução. O detalhamento da metodologia adotada pode ser consultado na Seção 4.1.2.

Em seguida, foi conduzida a etapa de *Implementação da Solução* por meio do Estudo de Caso 2, com quatro profissionais de uma equipe ágil. Os resultados deste estudo foram avaliados na etapa de *Avaliação da Implementação* deste ciclo, pois não houve um terceiro ciclo. Por fim, foi gerada a versão final do artefato. A metodologia aplicada nas etapas de *Implementação* e *Avaliação da Implementação* é discutida na Seção 4.1.2.

4.1.1 Metodologia aplicada no Ciclo 1

A seguir são detalhados os procedimentos metodológicos adotados nas etapas que compõem o Ciclo 1 desta DSR. Desse modo, são discutidas e apresentadas as metodologias aplicadas nas etapas de *Investigação do Problema*, *Design da Solução*, *Validação da Solução* e *Implementação da Solução*.

Investigação do Problema

A etapa de *Investigação do Problema* consiste em conhecer o problema abordado na pesquisa ou uma classe de problemas [78]. Esta etapa é caracterizada pela busca de informação, como uma forma de entender o problema que será solucionado nas etapas posteriores. Para isso, é importante definir os *stakeholders* afetados pelo problema, as causas que originam o problema, os objetivos a serem alcançados, os efeitos e as contribuições geradas a partir da solução do problema.

O problema investigado nesta pesquisa consiste em analisar como Testes Exploratórios podem ser aplicados em contextos ágeis, de projetos de desenvolvimento de *software* que possuem um cenário de requisitos que não são formalmente documentados e que mudam constantemente, e onde o cliente possui uma participação ativa no projeto. Os principais envolvidos e afetados pelo problema são os profissionais que integram equipes ágeis, como desenvolvedores, gerentes/líderes de projeto, dentre outros profissionais com o perfil multifuncional - característica do método ágil. O fenômeno investigado é o Teste Exploratório. A causa que origina este problema é a ausência de uma abordagem simples e robusta que viabilize a execução de ET no contexto prático e multifuncional das equipes ágeis.

De um modo geral, a contribuição desta pesquisa envolve a mitigação das limitações existentes para aplicação de ET em um contexto ágil, viabilizando o aumento da qualidade do produto desenvolvido. Alguns outros benefícios mais específicos podem ser percebidos como, a redução de custos no projeto, a economia de tempo no planejamento de *scripts* de teste, a detecção rápida de falhas no sistema, os testes registrados com mais eficiência, a diminuição de artefatos de teste e uma melhor compreensão do funcionamento do produto, pelo testador.

Portanto, para compreender o contexto do problema, foram conduzidas uma RSL, uma pesquisa por ferramentas de ET e entrevistas com profissionais ágeis. Os resultados obtidos culminaram em um curso de ET, destinado a profissionais ágeis, para validar a aplicação prática da abordagem SBTM (o Apêndice D apresenta o protocolo adotado para a implementação deste estudo). Os resultados desta etapa são apresentados na Seção 4.2.1.

Design da Solução

A etapa de *Design da Solução* consiste na construção de uma solução para resolver o problema identificado na etapa de *Investigação do Problema*. Wieringa [78] enfatiza que a solução gerada a partir de uma DSR pode ser proposta por meio da descrição de linguagem natural, diagramas, modelos matemáticos, protótipos, pela combinação destes, ou outros. Nesta etapa os requisitos são identificados de modo a perceber como eles podem atender aos objetivos da pesquisa.

Para a realização da etapa de *Design da Solução* do Ciclo 1 foi necessário, primeiramente, acompanhar a aplicação dos ET em projetos reais desenvolvidos por profissionais ágeis. Em seguida, foram coletados mais *insights* por meio de um *brainstorming* com esses profissionais. Essas abordagens foram necessárias para compreender os requisitos fundamentais para o desenho da solução. A Seção 4.2.2 apresenta os resultados gerados com a execução desta etapa.

Validação da Solução

Esta etapa é baseada na construção do conhecimento sobre a solução proposta, onde o pesquisador examina a solução a partir de um uso inicial no contexto do problema investigado [78]. Desse modo, esta etapa de *Validação da Solução* foi conduzida nos Ciclos 1 e 2, por meio de um estudo de validação empírica com profissionais da área, com o objetivo de verificar se a solução satisfaz os requisitos e atendeu aos objetivos esperados pelos *stakeholders*.

No Ciclo 1, a etapa de *Validação da Solução* teve como objetivo obter um *feedback* inicial sobre o método *Agile ETeasy*. Assim, a validação foi conduzida por meio de uma pesquisa de opinião com profissionais de equipes ágeis, que ainda não haviam participado deste estudo. A pesquisa de opinião foi conduzida por meio de um questionário *online*. Desse modo, os profissionais deveriam fornecer sua opinião sobre o entendimento e a utilidade do método *Agile ETeasy*, bem como sugestões de melhoria e críticas.

No Ciclo 1, a validação contou com a participação de cinco profissionais. Estes profissionais foram selecionados por conveniência, conforme conhecimentos sobre o tema abordado no método e disponibilidade em participar da pesquisa. O método *Agile ETeasy* foi submetido à análise de profissionais, individualmente e separadamente - de modo que

suas opiniões não influenciasses uns aos outros. Para isto, o método foi detalhado em um *website* para que a leitura e interpretação estivesse tão clara a ponto de não ser necessária uma explicação (por terceiros) sobre sua utilidade. Então, os profissionais foram estimulados a imaginar como o método iria interagir com o contexto imaginado por eles e, então, prever os possíveis efeitos (contribuições, dificuldades ou dúvidas).

Em síntese, na validação, os profissionais foram responsáveis por “observar” a forma como o método seria executado em projetos reais de desenvolvimento de *software*. E, em seguida, direcionados à responder um questionário *online* com 12 questões subjetivas e 17 questões objetivas sobre suas impressões, sugestões ou críticas sobre o método.

Nesse questionário os profissionais foram direcionados a informar: dados pessoais, que os caracterizam enquanto profissionais da área - como grau de instrução acadêmica, experiência profissional, dentre outros; aspectos que caracterizam a facilidade de entendimento do método *Agile ETeasy*; e, aspectos que caracterizam a utilidade do método *Agile ETeasy* no ambiente de trabalho. As questões abordadas neste questionário podem ser vistas no Apêndice E. Os dados coletados foram analisados a partir de uma análise qualitativa e os resultados são apresentados na Seção 4.2.3.

Implementação da Solução

A etapa de *Implementação da Solução* possui um caráter prático, ou seja, a solução elaborada é avaliada em um contexto real de uso, com o intuito de analisar sua facilidade de entendimento e de uso por profissionais da área [78]. No Ciclo 1, esta etapa foi planejada e conduzida após o *feedback* obtido por meio dos avaliadores na etapa de Validação da Solução. Assim, as sugestões de melhorias e as críticas apresentadas pelos avaliadores contribuíram para o refinamento do método *Agile ETeasy*, antes de ser aplicado no cotidiano de uma equipe ágil.

Para isso, foi selecionada uma equipe com perfil compatível ao contexto do problema investigado nesta pesquisa. Assim, foi aplicado um Estudo de Caso em um projeto de desenvolvimento de sistemas embarcados, com uma equipe composta por três profissionais (desenvolvedores). O projeto e a equipe foram selecionados por conveniência, conforme o *status* das *Sprints* do projeto e a disponibilidade para participar do estudo. Ou seja, nenhum critério estatístico foi usado para selecionar os profissionais. É importante destacar que a

equipe atua em um projeto de colaboração entre a academia e a indústria, e integra o *Software Practices Laboratory* (SPlab) da Universidade Federal de Campina Grande (UFCG).

Assim, para a implementação do *Agile ETeasy*, no Ciclo 1 e no Ciclo 2, foi elaborado um único protocolo (consultar o Apêndice G) para os dois Estudos de Caso, de acordo com Yin [80]. Este protocolo contém uma visão geral do projeto de Estudo de Caso, o *Design* do Estudo de Caso, a descrição da preparação, a coleta e a análise de dados.

Inicialmente, o Estudo de Caso foi conduzido por meio de (i) um questionário *online* semi-estruturado aplicado com os profissionais, com o intuito de coletar informações sobre a caracterização dos participantes e sobre sua experiência com Teste de *Software*. No total foram abordadas cinco questões subjetivas e três questões objetivas. Em seguida, foi conduzido (ii) um minicurso explicativo do método *Agile ETeasy*, para informar detalhes do método aos participantes. Após isso, os participantes foram orientados a (iii) executar o método em um módulo do sistema desenvolvido pela equipe. É importante destacar que o Estudo de Caso foi realizado no formato não-presencial, em virtude da pandemia ocasionada pelo *COrona VIRus Disease* (Doença do Coronavírus) ou *Sars-CoV-2*, que acomete a população brasileira, especificamente, desde fevereiro de 2020. O Apêndice H contém o questionário inicial aplicado com os participantes.

4.1.2 Metodologia aplicada no Ciclo 2

A seguir são detalhados os procedimentos metodológicos adotados nas etapas que compõem o Ciclo 2 desta DSR. Desse modo, são discutidas e apresentadas as metodologias aplicadas nas etapas de *Avaliação da Implementação*, *Design da Solução*, *Validação da Solução* e *Implementação da Solução*.

Avaliação da Implementação

A etapa de *Avaliação da Implementação* se refere à avaliação dos resultados obtidos a partir do Estudo de Caso (realizado na etapa anterior). O intuito desta etapa é avaliar se a solução consegue solucionar o problema identificado na etapa de *Investigação do Problema* [78], de forma prática. Assim, esta etapa de avaliação foi conduzida duas vezes: uma, para cada Estudo de Caso aplicado, na etapa de *Implementação da Solução*.

Como ambos os Estudos de Caso foram realizados no formato não-presencial, a execução do método *Agile ETeasy*, pelos participantes, não pôde ser observada na íntegra. Por este motivo, foi adotado um instrumento para fins de registro e acompanhamento das atividades executadas no estudo. Este instrumento é conhecido como Diário de Bordo. Os participantes foram motivados a preencher um Diário de Bordo *online* detalhando sua experiência na utilização dos elementos do método *Agile ETeasy*, pontos positivos, resultados alcançados, pontos negativos e/ou dificuldades encontradas. O Apêndice I apresenta o *template* com as orientações e os campos a serem preenchidos no Diário de Bordo desta pesquisa.

Após a implementação do método *Agile ETeasy*, os participantes foram entrevistados individualmente. Desse modo, foi possível que informações fornecidas no Diário de Bordo pudessem ser esclarecidas ou discutidas durante a entrevista. Informações que não foram possíveis de ser transcritas no Diário de Bordo também puderam ser expostas. Para conduzir as entrevistas foi elaborado um roteiro com questionamentos e assertivas sobre o entendimento e a utilidade do método *Agile ETeasy*. O Apêndice J apresenta o roteiro adotado nas entrevistas.

Por fim, os resultados coletados a partir do Diário de Bordo e das entrevistas com os participantes foram sintetizados e analisados a partir de uma análise qualitativa, baseada no *Grounded Theory* [21]. Para isso, usamos a codificação aberta, axial e seletiva. A Seção 4.3.1 apresenta os resultados desta avaliação.

Design da Solução

A etapa de *Design da Solução* no Ciclo 2 foi realizada após avaliação dos Estudos de Caso 1 e 2, e também após a segunda validação com os profissionais especialistas. Assim, o artefato gerado no Ciclo 1 foi refinado sucessivas vezes, no Ciclo 2, até chegar em sua versão final. Os resultados inerentes à esta etapa no Ciclo 2 são discutidos e apresentados na Seção 4.3.2. O método *Agile ETeasy* é descrito em detalhes no Capítulo 3.

É importante destacar que após cada validação com profissionais especialistas, a etapa de *Design da Solução* foi repetida, de modo a refinar este artefato.

Validação da Solução

No Ciclo 2, a etapa de *Validação da Solução* ocorreu a partir de alguns refinamentos já aplicados ao método *Agile ETeasy*, após a realização do Estudo de Caso 1. Neste sentido, a validação também foi conduzida por meio de uma pesquisa de opinião com profissionais especialistas (avaliadores) em desenvolvimento de *software* ágil, por meio de um questionário *online*. Do mesmo modo que na primeira *Validação*, os avaliadores deveriam fornecer sua opinião sobre o entendimento e a utilidade do método *Agile ETeasy*, bem como sugestões de melhoria e críticas.

Para isso, a validação contou com a participação de doze avaliadores. Estes profissionais foram selecionados por meio do método de amostragem não probabilística *snowball sampling*. Para isso, foram enviados *e-mails* para profissionais próximos e, em seguida, ampliou-se o envio para profissionais indicados por terceiros. No total, foram enviados 51 *emails* convidando os profissionais a participarem da pesquisa. No total, 12 profissionais participaram da validação do método *Agile ETeasy* por meio das respostas ao questionário. Embora, o questionário adotado nesta segunda validação tenha objetivos semelhantes ao questionário da primeira validação, algumas melhorias foram aplicadas à descrição dos questionamentos, visando simplificar a obtenção das respostas e, conseqüentemente, obter um alcance mais abrangente dos profissionais. Desse modo, o questionário aplicado na segunda validação pode ser visto no Apêndice F. E, finalmente, os resultados foram analisados de forma qualitativa e são discutidos na Seção 4.3.3.

Implementação da Solução

Para *Implementação da Solução* no Ciclo 2 desta DSR, foi selecionada uma equipe com perfil compatível ao contexto do problema investigado nesta pesquisa. Já no Ciclo 2, um segundo Estudo de Caso foi conduzido com uma equipe que atua com desenvolvimento *web*, composta por quatro profissionais (três desenvolvedores e um líder da equipe). Assim como no Ciclo 1, neste ciclo, tanto o projeto como a equipe foram selecionados por conveniência, conforme o *status* das *Sprints* do projeto e a disponibilidade para participar do estudo. Ou seja, nenhum critério estatístico foi usado para selecionar os profissionais. A equipe atua em projetos de colaboração entre a academia e a indústria, e integra o *Software Practices*

Laboratory (SPlab) da Universidade Federal de Campina Grande (UFCG).

Para a implementação do *Agile ETeasy*, no Ciclo 2, foi adotado o mesmo protocolo usado na etapa de *Implementação da Solução* no Ciclo 1 (consultar o Apêndice G), de acordo com Yin [80]. Este protocolo contém uma visão geral do projeto de Estudo de Caso, a descrição dos procedimentos e regras adotadas, das questões investigadas no Estudo de Caso e orientação para o relatório de dados do estudo.

Em síntese, o Estudo de Caso 2 também foi conduzido, por meio de (i) um questionário inicial aplicado com os profissionais, com o intuito de coletar informações sobre a caracterização dos participantes e sobre sua experiência com Teste de *Software*. No total foram abordadas cinco questões subjetivas e três questões objetivas. Em seguida, foi conduzido (ii) um minicurso explicativo do método *Agile ETeasy* para informar detalhes do método aos participantes. Após isso, os participantes foram orientados a (iii) executar o método em um módulo do sistema desenvolvido pela equipe.

É importante destacar que ambos os Estudos de Caso foram realizados no formato não-presencial, em virtude da pandemia ocasionada pelo *COrona VIRUS Disease* (Doença do Coronavírus) ou *Sars-CoV-2*, como justificado anteriormente. O Apêndice H contém o questionário inicial aplicado com os participantes.

Por fim, os resultados do Diário de Bordo e das entrevistas foram coletados, sintetizados e analisados a partir da codificação aberta, axial e seletiva orientadas na *Grounded Theory* [21]. A Seção 4.3.4 apresenta os resultados desta avaliação.

4.2 Ciclo 1 da DSR

Neste capítulo são apresentados os resultados das etapas de *Investigação do Problema*, *Design da Solução*, *Validação da Implementação* e *Avaliação da Implementação* do Ciclo 1 da *Design Science Research*, de acordo com os procedimentos metodológicos descritos na Seção 4.1.1.

4.2.1 Investigação do Problema de Pesquisa

Desse modo, com o intuito de aprender mais sobre o problema a ser investigado nesta pesquisa, esta etapa abordou o desenvolvimento de quatro atividades distintas: (1) a realização

de uma pesquisa bibliográfica, seguindo as orientações de Kitchenham e Charters [50] para a condução de uma RSL, que objetivou compreender quais as abordagens, técnicas ou modelos têm sido usados para apoiar a prática de ET no ambiente ágil, conforme descrito na Seção 4.2.1; (2) uma pesquisa por ferramentas de ET, com o intuito de verificar a existência de ferramentas que apoiam e contribuem com a execução de ET, conforme apresentado na Seção 4.2.1; (3) entrevistas com profissionais que integram equipes ágeis, para compreender o contexto da execução de práticas de ST e perceber como ET são adotados no cotidiano destas equipes, conforme discutido na Seção 4.2.1; e, (4) a elaboração de um curso sobre ET para profissionais de equipes ágeis, conforme descrito na Seção 4.2.1. Em seguida, uma síntese das contribuições geradas a partir da execução desta etapa é apresentada e discutida na Seção 4.2.1. E, por fim, as conclusões deste capítulo são expostas na Seção 4.4.

Pesquisa Bibliográfica (RSL)

De acordo com Kitchenham e Charters [50], o desenvolvimento de uma Revisão Sistemática da Literatura (RSL) deve seguir passos metodologicamente bem definidos, englobando três fases: o planejamento, a condução e a publicação de pesquisa. Assim, esta RSL tem como intuito realizar uma busca por estudos primários sobre ET desenvolvidos no contexto ágil. Para isso, é proposta a análise dos estudos primários a partir de critérios de exclusão e inclusão, critérios de qualidade, e, por fim, respostas à questões de pesquisa. O protocolo adotado nesta RSL é apresentado no Apêndice A.

Existem dois métodos principais de síntese de dados: descritivos (qualitativos) e quantitativos [50]. Desse modo, os dados extraídos foram analisados usando um método qualitativo para responder às questões de pesquisa, o que direcionou a uma síntese de dados descritivos. Para a síntese foi utilizada a análise de conteúdo para estruturar os resultados. Assim, como respostas para cada Questão Específica (QE) de pesquisa são apresentados sínteses e resumos narrativos. Portanto, os dados extraídos dos 17 trabalhos primários foram classificados e categorizados¹. Os resultados gerais são apresentados a seguir. Mais detalhes da pesquisa podem ser acompanhados em <https://bitly.com/hXyTThdI>.

¹Síntese geral dos dados: <https://bitly.com/uJdPfJcL>

QE1: *Quais estratégias de gerenciamento de ET têm sido usadas para apoiar a execução de ET no ambiente ágil?*

Para responder esta QE1 de pesquisa foram investigadas abordagens, técnicas, artefatos e ferramentas indicadas nos trabalhos primários levantados na RSL. Dentre as abordagens usadas no contexto de equipes ágeis, os trabalhos primários indicaram o uso de algumas abordagens mais comuns, como: *Session Based Test Management* (SBTM) [23]; e, *Team Exploratory Testing Sessions* (TET) [65].

Outras abordagens mais específicas também são relatadas nos trabalhos primários, como o uso de um *framework* para automação do ET [9; 81]; *checklist* para o *design* de Cartas de Teste [36]; e, um Mapa de Oportunidades (*Opportunity Map* - OM) [20].

Alguns artefatos também foram referenciados, como o plano de teste e relatórios com resultados de execuções de teste anteriores [22]. Bem como, o uso de técnicas para a aplicação do ET, como *Kanban*, registro das falhas em relatórios manuais ou automatizado (em forma de *logs* de teste) [22].

A literatura recente indica o uso de algumas ferramentas para executar e/ou gerenciar ET no contexto ágil. Alguns trabalhos destacam as ferramentas *Test Tracker* [53], *Xray Exploratory App* [23], e *Mantis*, *TestLink* e *Jira* e uso de planilhas [22].

QE2: *Quais os benefícios e de aplicar ET em projetos de desenvolvimento de software no contexto ágil?*

Para responder esta QE2 de pesquisa foram investigados os principais benefícios em aplicar ET no contexto ágil, relatados nos trabalhos primários. A seguir os benefícios identificados são listados.

- A implementação de ET nas Pequenas e Médias Empresas de *software* (PMEs) aumenta a eficiência do processo de identificação de *bugs* [9].
- As sessões TET podem ser uma forma útil de aplicar ET, para complementar outros tipos testes. E, podem ser usados para encontrar defeitos de usabilidade no *software* testado [65].
- Ganho significativo de produtividade pelos profissionais que aplicaram ET [35].

- Abordagens estruturadas de ET têm potencial para mitigar os desafios de testar funcionalidades que oferecem “muitos caminhos” [5].
- Reduz custos no projeto e economiza tempo, é capaz de detectar as vulnerabilidades no sistema, proporciona uma melhor compreensão de domínio do produto, pelo testador [10].

QE3: *Quais as limitações de aplicar ET em projetos de desenvolvimento de software no contexto ágil?*

Para responder esta QE3 de pesquisa, foram investigadas as principais limitações em aplicar ET no contexto ágil, relatados nos trabalhos primários. A seguir as limitações identificadas são listadas.

- Os profissionais de teste precisam ser melhor orientados na exploração dos sistemas e na identificação de *bugs* em cenários relacionados a dispositivos móveis [70].
- A eficiência do ET é significativamente afetada pela experiência do profissional que aplica o teste, enquanto que a formação educacional não tem influência sobre esse fator. Ou seja, o número de falhas críticas detectadas é maior quando o profissional é mais experiente [35].
- No geral, o ET ainda é abordagem de teste ágil informal [22].
- A ausência de uma abordagem bem definida e que seja compatível com o real contexto de trabalho das equipes ágeis. Nem sempre o que é proposto na literatura é aplicável, em sua completude, no contexto real dos profissionais ágeis [23].
- Um apoio mais específico para o planejamento e execução do ET, como orientações, ferramentas e artefatos de teste [22; 23].

Alguns estudos indicam que trabalhos adicionais para complementar o conhecimento da área de ET, no contexto ágil, também precisam ser realizados. Martensson et al. [55] afirma que a construção de um método ou modelo, pode ajudar as empresas a melhorar a forma como os ET são usados, bem como uma análise entre ET e métodos ágeis se faz necessária para *insights* mais específicos da área. Raappana et al. [65] também indica que

mais pesquisas são necessárias para estudar os efeitos dos ET em outros contextos de projetos e equipes ágeis, de modo a investigar as diferenças entre os resultados das sessões TET em comparação com outros tipos de testes. Gebizli et al. [35] reforça a necessidade de aplicar estudos experimentais para avaliar o impacto de diferentes fatores sobre a eficácia do ET.

Em linhas gerais, os resultados indicam que (i) ET ainda é uma área de estudo pouco investigada, quando se refere ao contexto ágil; (ii) a literatura recente indica poucas estratégias (abordagens e técnicas) de ET usadas na prática; (iii) no contexto prático, há uma limitação de ferramentas específicas para a aplicação de ET e que considere suas principais características; (iv) estudos empíricos com equipes ágeis são necessários para validar a aplicação prática de algumas abordagens e ferramentas existentes.

Pesquisa por ferramentas de ET

Com o objetivo de reduzir alguns dos problemas em executar Testes Exploratórios, como a dificuldade de medir a cobertura dos testes, de reproduzir os testes e gerar artefatos de testes (como relatórios, por exemplo) foi conduzida uma busca e uma análise por ferramentas que apoiam a execução dos ET. Ao todo foram encontradas dezoito (18) ferramentas ². Informações sobre o acesso às ferramentas podem ser consultadas no Apêndice B.

Inicialmente, cada ferramenta foi classificada de acordo com critérios pré-definidos, como disponibilidade de acesso da ferramenta (*Freeware* ou *Trial*) e suporte à plataforma *Web*, *Desktop* ou *Mobile*. As ferramentas classificadas como “*Freeware*” se referem aquelas que não apresentam custo para utilização e como “*Trial*” as ferramentas que oferecem o uso do produto como um teste ou um serviço por um período de tempo determinado. A Tabela 4.1 destaca com o símbolo “✓” as ferramentas que apresentam tais critérios.

Assim, as ferramentas categorizadas como “*Freeware*” foram selecionadas e analisadas de modo mais específico, a partir de funcionalidades direcionadas ao gerenciamento dos ET. Do total de 18 ferramentas classificadas, parte delas (11) são ferramentas que dispõem de uma versão “*Trial*”, ou seja, após um período de tempo solicitam o pagamento por sua licença de uso. E, por este motivo, podem ou não disponibilizar todas as suas funcionalidades. Portanto, apenas 07 ferramentas puderam ser analisadas em relação à suas funcionalidades.

A análise de funcionalidades aplicada às ferramentas “*Freeware*” foi embasada no

²Descrição geral das ferramentas de ET: <https://bitly.com/bsXLJXaK>

suporte que a ferramenta disponibiliza para o gerenciamento dos ET. Assim, foram consideradas Funcionalidades (F) referentes a: *report textual* (F1); *report textual* e captura de tela (F2); *report textual*, captura de tela e vídeos (F3); *capture replay* (F4); gravação de áudio (F5); permite anexos ao *report* e/ou ao planejamento (F6); permite importar planos de testes/missões (F7); permite exportar planos de testes/missões (F8); geração/injeção de dados (F9); cronômetro (F10); integração ao *Jira Software*³ (F11); constrói o plano de teste/missão (F12); agenda atividades de teste (F13); permite a triagem de problemas e defeitos (F14); e, gerencia os requisitos (F15). A Tabela 4.2 apresenta a análise destas funcionalidades em cada ferramenta, destacando com o símbolo “✓” aquelas que as apresentam.

Na análise de funcionalidades, apresentadas na Tabela 4.2, foram analisadas em suas versões *web* as ferramentas *Bug Magnet*, *Exploratory Testing Chrome Extension*, *QMetry Exploratory Testing* e *Spira Capture*; e, em suas versões *desktop*, as ferramentas *Rapid Reporter*, *Wink* e *XRay Exploratory App*. Nenhuma versão *mobile* foi analisada, pois as equipes ágeis que contribuiriam com este estudo não desenvolvem projetos *mobile*. Neste caso, este tipo de ferramenta não se aplicaria ao contexto de estudo. É importante destacar que as funcionalidades analisadas nas ferramentas não se limitam, apenas, as que são aqui apresentadas e que profissionais ágeis ou testadores podem utilizar estas ferramentas de diversas formas, de acordo com necessidade do projeto ou o contexto de trabalho.

Após esta análise, apenas duas ferramentas demonstraram ser mais eficientes para aplicação de ET: a ferramenta *Spira Capture*⁴ e a ferramenta *Xray Exploratory Testing App*⁵. A ferramenta *Spira Capture* se refere à uma extensão para navegador *web*. Esta extensão registra e organiza as sessões de ET, permitindo que as gravações de texto, de tela e de vídeo sejam salvas.

³*Jira Software*: <https://www.atlassian.com/br/software/jira?bundle=jira-software&edition=free>

⁴*Spira Capture*: <https://www.inflectra.com/SpiraCapture/>

⁵*Xray Exploratory Testing App*: <https://www.getxray.app/exploratory-testing>

Tabela 4.1: Análise das ferramentas em relação a disponibilidade de acesso e suporte à plataforma.

| Ferramenta | Disponibilidade da ferramenta | | Suporte à plataforma | | |
|---|--------------------------------------|--------------|-----------------------------|----------------|---------------|
| | <i>Freeware</i> | <i>Trial</i> | <i>Web</i> | <i>Desktop</i> | <i>Mobile</i> |
| <i>Azure Test Plans</i> | | ✓ | ✓ | ✓ | |
| <i>BB Assistant</i> | | ✓ | ✓ | ✓ | |
| <i>Bug Magnet</i> | ✓ | | ✓ | | |
| <i>Exploratory Testing Chrome Extension</i> | ✓ | | ✓ | ✓ | ✓ |
| <i>Jira + Jira Capture</i> | | ✓ | ✓ | | |
| <i>Practi Test</i> | | ✓ | ✓ | ✓ | ✓ |
| <i>QMetry Exploratory Testing</i> | ✓ | | ✓ | ✓ | ✓ |
| <i>Qtest Explorer</i> | | ✓ | ✓ | ✓ | |
| <i>Rapid Reporter</i> | ✓ | | ✓ | ✓ | ✓ |
| <i>Rapise</i> | | ✓ | ✓ | ✓ | ✓ |
| <i>SmartBear Zephyr</i> | | ✓ | ✓ | ✓ | |
| <i>Spira Capture</i> | ✓ | | ✓ | ✓ | |
| <i>Testpad</i> | | ✓ | ✓ | ✓ | ✓ |
| <i>Testrail</i> | | ✓ | ✓ | ✓ | ✓ |
| <i>Test Studio Explorer</i> | | ✓ | ✓ | ✓ | ✓ |
| <i>Teststuff</i> | | ✓ | ✓ | ✓ | ✓ |
| <i>Wink</i> | ✓ | | ✓ | ✓ | ✓ |
| <i>XRay Exploratory App</i> | ✓ | | | ✓ | ✓ |

Tabela 4.2: Análise das ferramentas em relação às funcionalidades.

| Ferramenta | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 | F11 | F12 | F13 | F14 | F15 |
|---|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| <i>Bug Magnet</i> | | | | | | | | | ✓ | | | | | | |
| <i>Exploratory Testing Chrome Extension</i> | ✓ | ✓ | | | ✓ | | | ✓ | | | | | | | |
| <i>QMetry Exploratory Testing</i> | ✓ | ✓ | ✓ | | | | | ✓ | | | ✓ | | | | |
| <i>Rapid Reporter</i> | ✓ | ✓ | | | | | | | ✓ | | | | | | |
| <i>Spira Capture</i> | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | | | |
| <i>Wink</i> | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | | | | |
| <i>XRay Exploratory App</i> | ✓ | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ |

A ferramenta *Spira Capture* também permite capturar como o usuário está interagindo com a guia atual do navegador por meio de comandos como cliques, *keystrokes*, tecla *enter*, mudanças na *url* ou erros de rede. Desse modo, é possível rastrear *bugs* e capturar os dados manualmente pelo testador, por meio de etapas, notas ou capturas de tela. É possível também, gerar relatórios a partir dos testes realizados. Em síntese, o *Spira Capture* busca auxiliar na documentação do ET e no *report* produzido.

A ferramenta *XRay Exploratory App* possui versões *desktop* e *mobile*. Em sua versão *Xray Test Management* é possível de integrá-lo ao *Jira Software*. Neste estudo, a integração desta ferramenta ao Jira não foi possível de ser realizada. O *XRay Exploratory App* auxilia na detecção de *bugs*, enquanto as sessões de ET são gravadas em formato de vídeo, áudio e/ou captura de tela. As sessões de ET são detalhadas e executadas diretamente na ferramenta. Ao encerrar a sessão, um relatório é anexado automaticamente à execução de teste - essa estratégia fornece um rápido *feedback* aos testadores. Em síntese, o *XRay Exploratory App* auxilia no *report* do teste e na documentação produzida.

Entrevistas com profissionais ágeis

Nesta etapa de *Investigação do Problema* também foram realizadas entrevistas com profissionais de equipes ágeis, para: **analisar** o contexto atual das atividades de Teste de Software praticadas nos projetos desenvolvidos por equipes ágeis, **com o propósito de** compreender como essas atividades são executadas, **em relação à** frequência dos testes aplicados nos projetos, ao tipo de teste executado, às ferramentas usadas, às dificuldades existentes e ao planejamento dos testes, **do ponto de vista de** cinco líderes de equipes ágeis, **no contexto de** projetos de desenvolvimento de *software* desenvolvidos pelo *Software Practices Laboratory* da Universidade Federal de Campina Grande (UFCG).

Para isso, foram definidas cinco Questões de Pesquisa (QP) que nortearam o planejamento do roteiro das entrevistas (consultar o Apêndice C). A seguir listamos as QPs deste estudo.

*QP*₁: Com qual frequência os testes são executados nos projetos?

*QP*₂: Qual o tipo de teste é mais comum de ser executado pela equipe?

*QP*₃: Qual ferramenta tem sido usada executar os testes? E de requisitos?

*QP*₄: Quais são as dificuldades de executar testes na equipe?

Tabela 4.3: Tempo de duração de cada entrevista

| Entrevistado | Tempo de duração |
|----------------|------------------|
| Entrevistado 1 | 00:31:55 |
| Entrevistado 2 | 00:39:42 |
| Entrevistado 3 | 00:25:15 |
| Entrevistado 4 | 00:31:39 |
| Entrevistado 5 | 00:39:58 |

QP₅: Como os testes são definidos ou planejados?

As entrevistas tiveram uma duração média de 25 à 39 minutos e foram realizadas com cinco líderes de equipes que atuam no contexto ágil (conforme apresentado na Tabela 4.3).

Todos os entrevistados possuem experiência com atividades de desenvolvimento de *software* e possuem formação acadêmica na área de Ciência da Computação (conforme apresentado na Tabela 4.4). Boa parte dos entrevistados iniciou sua experiência profissional em estágios oportunizados pela UFCG ou como *freelancers*.

Tabela 4.4: Caracterização dos entrevistados.

| Entrevistado | Perfil do entrevistado | Formação acadêmica |
|----------------|---|---|
| Entrevistado 1 | Desenvolvedor <i>web</i> e Líder de equipe | Graduação em Ciência da Computação - concluído |
| Entrevistado 2 | Desenvolvedor <i>embedded</i> e Líder de equipe | Doutorado em Ciência da Computação - em andamento |
| Entrevistado 3 | Desenvolvedor <i>web</i> e Líder de equipe | Graduação em Ciência da Computação - concluído |
| Entrevistado 4 | Desenvolvedor <i>web</i> e Líder de equipe | Especialização em Arquitetura de <i>Software</i> - em andamento |
| Entrevistado 5 | Desenvolvedor <i>web</i> e Líder de equipe | Graduação em Ciência da Computação - concluído |

Sobre o método ágil adotado nas equipes, os entrevistados afirmaram, em unanimidade,

que adotam ou tentam adaptar as práticas do SCRUM ao contexto dos seus projetos. Entretanto, o entrevistado 5 ressaltou que “*Lá no projeto a gente tenta usar o SCRUM. Mas, nem sempre a gente consegue seguir fielmente todas as etapas do processo. Mas, isso se dá por conta da dinâmica que foi estabelecida com o cliente. Dependendo das necessidades do cliente, a gente foi adaptando um pouco esse processo...*”.

Em relação à frequência da execução de testes nos projetos gerenciados pelos líderes entrevistados (QP_1) foi constatado que os testes geralmente são realizados no final das entregas do projeto, quando o desenvolvedor submete uma *task* no JIRA. Em alguns casos, os testes são feitos a medida que o código vai sendo revisado ou quando um *bug* é detectado. Em outros casos, os testes ocorrem desde o início da implementação das *task*.

Foi informado pelos entrevistados que o tipo de teste mais comum nos projetos (QP_2) são os testes funcionais. Para isto, é gerado e testado um *script* simples. Em outros casos, são feitos testes manuais ou unitários, pela facilidade de desenvolver esse tipo de teste.

Sobre a ferramenta mais usada para executar os testes (QP_3), destacou-se uma Biblioteca do *Java*, o JEST; e, para o gerenciamento de requisitos, o JIRA. Um dos entrevistados ainda citou o *framework Mocha*, para testes em um sistema *desktop*, devido a sua flexibilidade e simplicidade.

Também foram investigadas as dificuldades dos entrevistados sobre a execução dos testes (QP_4). A Tabela 4.5 destaca os depoimentos dos entrevistados sobre suas principais dificuldades.

Em relação ao planejamento dos testes (QP_5), os entrevistados citaram que, em alguns casos, “*é criado uma ‘tabelinha’ (planilha)*” (Entrevistado 1) para o detalhamento do caso de teste. Em outros casos, “*os testes vão sendo executados desde o início da implementação das task*” (Entrevistado 2), mas não há um planejamento específico para o teste. Foi destacado também que “*os testes são feitos mais sobre demanda e não há uma pessoa específica - quem estiver com pouca demanda, testa naquela momento*” (Entrevistado 4). De acordo com o Entrevistado 5, ainda não há, na equipe, “*uma forma muito madura de desenvolver os teste*”.

No geral, constatamos que (i) os participantes tem pouca demanda e pouco tempo para testar; (ii) testes manuais ainda são muito úteis, embora demandem muito tempo de dedicação e seja uma tarefa repetitiva; (iii) boa parte dos participantes não tiveram uma formação

Tabela 4.5: Depoimentos dos entrevistados sobre as dificuldades em executar testes nos projetos.

| Entrevistado | Depoimento |
|---------------------|---|
| Entrevistado 1 | <i>“Como não é algo estabelecido para os desenvolvedores, acaba que as vezes não é feito! Ou a gente acaba desenvolvendo testes manuais, que demoram mais... eu diria que isso prejudica um pouco.”</i> |
| Entrevistado 2 | <i>“Um problema é o fato de não termos testes automáticos. Os testes que executamos manualmente, não tem tanto problema, porque a gente consegue executar ‘direito’. (...) Os testes automáticos podem ser de grande ajuda, mas não ‘serão a bala de prata’ do projeto, pois muitas coisas precisam ser testadas com testes manuais.”</i> |
| Entrevistado 3 | <i>“Acaba faltando tempo suficiente pra testar. Em termos de formação complementar, só cursei uma disciplina de teste, sem muitos detalhes.”</i> |
| Entrevistado 4 | <i>“A dificuldade talvez venha da parte de, se a tarefa for especificada de maneira equivocada, conseqüentemente o código vai ser executado de forma equivocada, e os testes também.”</i> |
| Entrevistado 5 | <i>“A gente não tem tanto contato com essa parte de teste, nem durante a graduação. Então, a gente é mais acostumado a estar desenvolvendo. (...) a gente também não tem tanta demanda pra teste, apesar de que deveria ter.”</i> |

específica em Teste de *Software*, aspecto que dificulta a aplicação de alguns testes mais específicos; (iv) os testes geralmente não são planejados; (v) na equipe, não há um profissional dedicado exclusivamente aos testes - característica do ágil; (vi) ET não é uma prática comum e é pouco conhecida entre os entrevistados; e, foi levantada (vii) a necessidade de uma estratégia simples e robusta para testar as aplicações.

Elaboração de um curso sobre ET

Diante das informações descobertas anteriormente e considerando os benefícios do ET no desenvolvimento ágil, foi percebida a necessidade de capacitar profissionais ágeis com base

na aplicação prática desse tipo de teste. Assim, foi conduzido um curso sobre ET com profissionais de desenvolvimento de *software* que atuam no contexto ágil e, em seguida, foi feito o acompanhamento da equipe em sessões de ET aplicadas em um projeto real (conforme descrito na Seção 4.2.2).

Para a realização do curso, foi selecionada uma abordagem de ET, o SBTM. Esta abordagem foi escolhida porque, dentre as demais (citadas na Seção 2.4) é a que mais se aproxima ao contexto das equipes ágeis, participantes deste estudo. Foi considerado, também, o fato de que o SBTM é usado como base para outras abordagens de ET, como o *Team Exploratory Testing Sessions* e o *Exploratory Testing in Pairs*. E, adicionalmente, a clareza, completude e o detalhamento de tarefas e artefatos - aspecto não percebido em outras abordagens de ET, como o *Tours* e o *Rule-Based Exploratory Testing*.

O Apêndice D descreve o planejamento do curso de ET, organizado em quatro etapas: definição do Plano de Curso, elaboração dos materiais de aula, elaboração das atividades práticas e elaboração do questionário de avaliação. Na Tabela 4.6 a estrutura do curso é apresentada, juntamente com a descrição dos tópicos e conteúdos abordados na ementa, e das atividades práticas planejadas para o final de cada módulo de aula. Adicionalmente, é informada a carga horária definida para cada módulo do curso. O material deste curso pode ser acessado em: <https://bitly.com/ENqikdPI>.

Após a aplicação do curso, as opiniões dos participantes foram coletadas e analisadas por meio de um formulário *online*. Ao total, foram consideradas as informações fornecidas pelos doze participantes do curso, pois todos concordaram em participar deste estudo, acompanharam as discussões durante as aulas e realizaram todas as atividades práticas previstas ao final de cada módulo do curso. Desse modo, esses resultados foram sintetizados⁶ e discutidos em Coutinho et al. [23].

Os resultados indicam a importância de um planejamento simples para a execução do ET; que artefatos de requisitos, mesmo pouco detalhados, podem contribuir o *Setup* da Sessão; a importância da Reunião de Alinhamento, como estratégia para registrar possíveis falhas, criar possíveis casos de testes formais, criar novas missões, registrar possíveis requisitos, e registrar novos *tests points*; a utilidade e importância de artefatos simples de ET, gerados na

⁶Síntese dos resultados do Questionário *online* aplicado no Curso de ET: <https://bitly.com/ocZKUTlr>

aplicação do SBTM, são úteis para a execução do ET; a relevância de definir Heurísticas em ET; dentre outros aspectos.

É importante destacar que (i) embora durante o curso, os participantes estivessem dispersos geograficamente, não foi abordado nenhum processo de Desenvolvimento Distribuído de *Software* (*Distributed Software Development - DSD*) em específico. A proposta do curso foi utilizar estratégias e ferramentas que viabilizassem ET no contexto de trabalho isolado e remoto; (ii) o uso de uma ferramenta específica (*Xray Exploratory App*) para o planejamento, execução e *report* de *bugs* (com gravação de vídeos, captura e anotações em *screenshots*, apenas anotações, dentre outros aspectos) em sessões de ET são benefícios não encontrados em outras ferramentas que auxiliam a execução de ET; (iii) a experiência do ensino remoto com participantes distribuídos geograficamente é desafiadora e fatores como o estímulo de habilidades de participação, colaboração e atenção precisam ser considerados para que a aprendizagem, de fato, aconteça; (iv) percebemos: o interesse e o engajamento dos participantes, ao praticarem o conteúdo teórico por meio de um problema real adotado nas atividades e nas discussões durante as aulas, em virtude, principalmente, do conhecimento pré-construído por meio do acesso prévio aos materiais da aula; a qualidade das respostas nos exercícios, pois boa parte dos artefatos de teste gerados foram de acordo com os critérios sugeridos na descrição das atividades; uma motivação em usar a ferramenta *Xray Exploratory App*, pela facilidade na criação, execução e exploração do ET realizados; dentre outros aspectos já discutidos nesta Seção.

Contribuições da Etapa de *Investigação do Problema*

Na etapa de *Investigação do Problema* foram conduzidos quatro estudos distintos com o objetivo de coletar informações que auxiliassem no entendimento do problema investigado, de modo a contribuir também com o levantamento de características fundamentais para o desenvolvimento do artefato gerado nesta tese.

Desse modo, compreender o contexto atual de ET relatados na literatura de *Agile ST*, bem como as práticas de teste executadas por profissionais que atuam no contexto ágil, as ferramentas que apoiam a execução de ET e as principais limitações de se aplicar uma abordagem específica de ET (como o SBTM), se fazem cruciais para a construção de uma proposta que atenda às reais necessidades de equipes que desenvolvem *software* por meio de práticas ágeis.

Tabela 4.6: Estrutura do Curso de Testes Exploratórios usando a abordagem SBTM

| Tópicos | Conteúdos | Atividade Prática (AP) | Carga Horária |
|-------------------------------------|--|--|---------------|
| Módulo I - Introdução | 1.1. O que são ETs? 1.1.1. Características de ETs 1.2. O que não são ETs? 1.2.1. Aleatoriedade e Testes <i>Ad Hoc</i> 1.2.2. Testes <i>Scripted</i> 1.3. Quando usar ET? | Objetivo da AP1: Compreender o propósito, criar hipóteses e planejar cenários de teste. | 02h |
| Módulo II - ETs na prática | 2.1. Heurísticas de ET 2.2. Planejamento de ET 2.3. Escrevendo Casos de ETs: <i>Charters</i> 2.4. Introdução a Testes Baseados em Sessão (SBTM) 2.5. Execução de testes baseado em sessões 2.6. Avaliação de uma sessão | Objetivo da AP2: Investigar Heurísticas, executar os testes e registrar as falhas. Objetivo da AP3: Aplicar as métricas de <i>Task Breakdown Structure</i> (TBS). | 04h |
| Módulo III - Um pouco mais sobre ET | 3.1. Problemas e Desafios, e Soluções 3.2. Boas práticas de ET 3.3. Ferramentas de Apoio aos ETs | Objetivo da AP4: Praticar o uso da ferramenta <i>Xray Exploratory App</i> por meio da execução de uma Sessão de ET. | 02h |

Assim, após a realização dos estudos aplicados na etapa de *Investigação do Problema*, foi possível definir o problema estudado nesta pesquisa. Em síntese, o problema investigado nesta pesquisa consiste em investigar estratégias para conduzir a execução de ET por equipes ágeis, que possuem um perfil multifuncional em um cenário de pouco tempo disponível para executar testes, de requisitos que não são formalmente documentados e que mudam constantemente, e onde o cliente possui uma participação ativa no projeto. Os principais *stakeholders* são os profissionais que integram equipes de desenvolvimento de *software* e adotam métodos ágeis em seu cotidiano de trabalho.

As causas que originam este problema são (i) a ausência de uma abordagem robusta e simples que oriente a aplicação de ET de forma prática, e (ii) a falta de um profissional experiente, dedicado exclusivamente a execução dos testes, que condiciona outro membro da equipe - geralmente, o desenvolvedor - a executar os testes em um cenário de *sprints* muito curtas. Por esses motivos, por vezes, as funcionalidades desenvolvidas pelas equipes são testadas de forma superficial ou nem chegam a ser testadas. E, adicionalmente, (iii) a indicação de artefatos gerados em ET não condizentes com a real necessidade das equipes ágeis, ou seja, artefatos obsoletos.

4.2.2 Design da Solução

Para o *Design da Solução* do artefato gerado nesta DSR foram considerados os resultados do acompanhamento das equipes ágeis na implementação da abordagem SBTM, por meio de um *brainstorming*, conforme apresentado na Seção 4.2.2. Em seguida, foi construída a primeira versão do método *Agile ETeasy* - conforme apresentado na Seção 4.2.2.

Acompanhamento das equipes ágeis

Após o curso de ET, houve o acompanhamento da implementação das práticas de SBTM, aprendidas e indicadas no curso. Este acompanhamento se deu em três momentos, logo após a finalização do curso, por meio de uma atividade prática em um cenário de um sistema *web*; e, por duas vezes, após alguns meses da aplicação do curso, em um cenário de um projeto real desenvolvido pelos participantes do curso, em seu cotidiano de trabalho.

Para realizar o acompanhamento das equipes na execução dos ET - no contexto prático

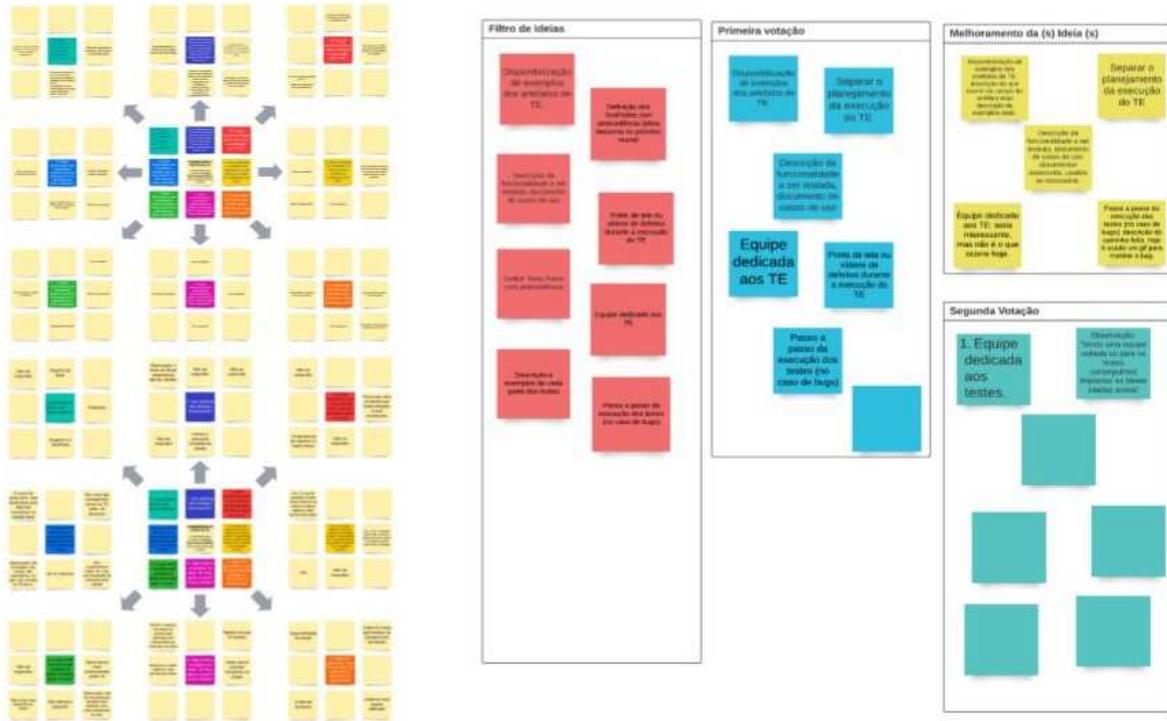


Figura 4.2: Execução do *Brainstorming*.

e real dos participantes - foi necessário aguardar o desenvolvimento de algumas *sprints* do projeto em que os mesmos atuam. Por este motivo, esta atividade foi executada por duas vezes - uma, em 20 de julho de 2021; e, outra, em 25 de janeiro de 2022.

No primeiro momento, o acompanhamento serviu para fornecer um apoio à implantação dos ET nas equipes. No segundo momento, houve o acompanhamento, seguido da análise e avaliação da abordagem SBTM. Por meio de um *brainstorming*, ideias foram levantadas sobre as características que favoreceram ou limitaram a execução dos ET por meio do SBTM.

Após algumas orientações iniciais, os participantes foram conduzidos às atividades do *brainstorming*. No total, o *brainstorming* durou um período de setenta e nove (79) minutos. É importante destacar que o *brainstorming* durou mais tempo do que o previsto no planejamento inicial. A Figura 4.2 ilustra a execução do *brainstorming* em *post its*, apresentado pela ferramenta *online Lucidspark*.

Foram conduzidas as seis atividades planejadas para o *brainstorming*. Não houve um limite máximo ou mínimo de ideias a serem expressas pelos participantes. Assim, os participantes foram estimulados a exporem suas ideias no intervalo de tempo definido para cada atividade (consultar Apêndice D).

Na *Atividade 1* foram obtidas 37 respostas aos questionamentos específicos elencados na preparação do *brainstorming*. Em seguida, na *Atividade 2*, alguns questionamentos foram feitos, de modo a esclarecer dúvidas relacionadas às ideias expostas. Na *Atividade 3*, algumas ideias precisaram ser agrupadas, e outras descartadas. Assim, um total de 29 ideias que não estavam alinhadas com o contexto principal do *brainstorming* foram desconsideradas. É importante destacar que, nem todas as respostas obtidas foram consideradas ideias viáveis de serem aplicadas, pois algumas estavam repetidas, complementavam umas as outras, ou estavam fora do contexto de pesquisa. Na *Atividade 4*, foram votadas 06 ideias a serem inseridas na próxima fase. Na *Atividade 5*, as ideias mais votadas foram discutidas brevemente e melhoradas, com o intuito de favorecer a votação a ser feita na próxima atividade. Por fim, a *Atividade 6* resultou em uma única ideia viável de ser implementada.

As ideias apresentadas no *brainstorming* foram relacionadas à implementação de ET no cotidiano dos participantes, a partir da aplicação do curso de ET. Assim, a exposição de algumas ideias foram determinantes para compreender a eficácia e a utilidade das práticas exercitadas e dos artefatos gerados. Categorizamos essas ideias para explicitar o que, de fato, se aplica e o que não se aplica ao cotidiano das equipes ágeis. A seguir é listada uma síntese das ideias expostas no *brainstorming* e que favoreceram a incorporação dos ET no cotidiano das equipes ágeis:

- O registro dos *Test Points* e o Relatório de Testes foram considerados importantes para o planejamento e a execução do ET.
- Em relação aos benefícios para o dia-a-dia da equipe, foi destacada a importância de registrar os ET feitos, no sentido de evidenciar os pontos que foram testados e suas pendências.
- O registro de *TestPoints* foi tido como uma contribuição significativa das Sessões de ET, pois na prática, houve uma melhoria na atividade de registro do teste a ser feito.
- Considerando o contexto de trabalho distribuído, adotar arquivos ou artefatos com permissão para colaboração simultânea, e ferramentas *online* contribuíram para o ET realizado remotamente.

Abaixo é listada uma síntese das ideias expostas no *brainstorming* que se destacaram como fatores limitantes à incorporação dos ET no cotidiano das equipes ágeis:

- O limite mínimo e máximo de tempo para execução de uma Sessão de ET não se aplica na prática, pois este fator é relativo à funcionalidade testada. Bem como, a gravação da execução completa de um sessão de ET.
- A indisponibilidade de tempo para planejar as sessões de ET foi definida como uma dificuldade na implantação do ET, na prática.
- Haver um processo de ET bem definido poderia contribuir com a inserção do ET como prática de teste no cotidiano da equipe.
- O tempo mínimo reservado para uma sessão de ET, no SBTM, poderia ser menor para funcionalidades pequenas.
- A falta de experiência com o ET faz com que o profissional que realiza o teste, dedique um esforço significativo para a preparação da sessão de ET como um todo.
- A ausência de um processo bem definido que otimize o tempo de preparação de uma sessão de ET limita a inserção do ET na equipe, bem como a organização e o empenho de todo a equipe para encontrar um momento em comum para realizar o ET.

De forma complementar, algumas outras ideias surgiram para facilitar a inserção dos ET no cotidiano das equipes, como:

- Incluir uma descrição e exemplos nos atributos de cada artefato da sessão de ET, para facilitar o entendimento do artefato, ou exemplificar a descrição de um *Test Point*.
- Definir os *Test Points* com antecedência.
- Um documento ou um modelo de *Use Case* poderia contribuir como artefato de requisitos base para o planejamento das sessões de ET.
- Disponibilizar uma breve descrição sobre a funcionalidade a ser testada, na sessão de ET, poderia facilitar o planejamento e a execução do teste.

- Capturar o *print screen* ou gravações da tela que contém os defeitos de funcionalidade identificadas na sessão de ET contribui com o registro do teste executado.
- Após a execução do teste, seria interessante registrar o passo a passo da execução do teste, o cenário testado e algum impedimento ou dificuldade encontrado pelo testador.

Em síntese, fatores como o planejamento dos *Test Points* e o registro dos testes executados foram características que favoreceram a incorporação do ET no cotidiano das equipes. Definir o grau de importância de cada *Test Point* foi caracterizado como um aspecto que facilitou o entendimento do que é prioridade de teste para o momento. Por outro lado, documentar o que foi testado também contribuiu com a condução da reunião de alinhamento com a equipe, considerando que todas as informações inerentes a execução do teste, como *bugs*, sugestões de melhorias ou pendências foram registradas.

Entretanto, os artefatos adotados nas sessões de ET se mostraram complexos ou com campos subutilizados para o registro do teste. Havendo, neste caso, a necessidade da implementação de orientações ou exemplos para esclarecer o uso eficaz do artefato. Foi constatada, portanto, a necessidade de adaptações nos artefatos - de planejamento e de registro do teste - para uma adequação mais específica ao contexto ágil em que o ET foi conduzido.

Algumas limitações na implementação das sessões de ET também foram percebidas, como: (i) a orientação de tempo mínimo e máximo para a realização das sessões: por vezes, o tempo mínimo (30 minutos, indicado pelo SBTM) para a sessão não foi utilizado porque a funcionalidade testada era muito simples, e a sessão que poderia ter sido feita menos tempo precisou ser estendida para atingir o tempo mínimo. Neste caso, uma nova orientação para situações com esta precisa ser considerada. Outra limitação se refere (ii) a ausência de um processo ou abordagem bem definida e que esteja compatível com o real contexto de trabalho das equipes ágeis. Nem sempre o que é proposto na literatura, como o SBTM, é aplicável em sua completude no contexto real dos profissionais ágeis. Foi percebido também uma limitação inerente à (iii) experiência do profissional que define os *Test Points*, pois quando esta atividade é executada por profissionais que desconhecem a aplicação a ser testada, existe o risco de se especificar um *Test Point* de forma inconsistente ou incompleta, gerando lacunas no planejamento do ET e gerando dificuldades na sua posterior execução.

Para incorporar, de fato, ET como prática de teste em equipes ágeis ainda se faz necessá-

rio definir um método que se adeque ao contexto destes profissionais, de modo a considerar orientações de aplicação prática, ferramentas mais específicas que considerem as particularidades do planejamento e da execução do ET, e artefatos simples, claros e eficazes de serem adotados. Por esse motivo, surge a necessidade de um método que se adeque às necessidades dos profissionais que atuam no desenvolvimento ágil e que vá além do conceito apresentado na literatura sobre ET.

Diante das informações descobertas nas entrevistas e considerando os benefícios do ET no desenvolvimento ágil (constatado na RSL), foi percebida a necessidade de integrar a teoria e a prática em ET para um melhor entendimento dos efeitos desse tipo de teste no ambiente ágil. Os resultados indicam que o SBTM, embora seja uma abordagem completa e robusta para a aplicação de ET, não se aplica ao contexto prático das equipes, em virtude da forma em que os artefatos de teste devem ser mantidos e do conhecimento prévio necessário para sua correta implementação.

De modo complementar, a ausência de orientações claras para guiar a execução do SBTM, dificulta que profissionais sem a *expertise* do ET consigam implementá-lo de forma eficaz, considerando, principalmente, os artefatos gerados. Por isso, é proposto nesta tese uma simplificação da abordagem SBTM, através da proposta de um método que otimiza o planejamento e a execução de ET em contextos ágeis, como exposto na seção seguinte.

Construção do Método *Agile ETeasy*

Considerando os artefatos que podem ser elaborados como solução de um problema na DSR, esta tese propõe **um método composto por etapas, atividades, tarefas, técnicas e elementos de saída, que visa auxiliar a implementação de Testes Exploratórios em contextos ágeis**. Este método é denominado *Agile ETeasy* (do inglês, *Agile Exploratory Testing Easy*).

O método *Agile ETeasy* é indicado para organizações e equipes que atuam com métodos ágeis, possuem um cliente participativo em seu projeto e possuem um cenário de requisitos que não são formalmente documentados e que mudam constantemente; como também, desejam inserir práticas de testes mais acessíveis ao seu cotidiano de trabalho, de modo a colaborar com a qualidade do *software* em desenvolvimento.

Dessa forma, o método *Agile ETeasy* foca na definição de etapas e fases que facilitam a operacionalização dos Testes Exploratórios por equipes ágeis. De forma mais prática, o



Figura 4.3: Estrutura geral do Método *Agile ETeasy*.

método *Agile ETeasy* tem como principais objetivos:

- Propiciar que Testes Exploratórios sejam realizados no contexto de equipes ágeis, de modo a facilitar a execução dos testes em um cenário de requisitos que não são formalmente documentados e que mudam constantemente, e cuja participação do cliente é ativa.
- Estabelecer uma estratégia de uso comum de Testes Exploratórios entre profissionais e pesquisadores da área.
- Alinhar a compreensão dos pesquisadores na discussão sobre a influência dos Testes Exploratórios durante o desenvolvimento de *software*.

De modo conceitual, o PMI [73] define processo como um conjunto de ações que levam a um resultado e é composto por: Entradas, Ferramentas e técnicas, e Saídas. Assim, para sistematizar o entendimento do método proposto, o *Agile ETeasy* foi categorizado em elementos que caracterizam um processo, ou seja, em: etapas, que organizam as partes do método; atividades e tarefas, que caracterizam as *Entradas* do método; técnicas, como elemento que se refere às *Ferramentas e Técnicas* do método; e, elementos de saída, que tratam das *Saídas* do método. As atividades se referem a um conjunto de tarefas que devem ser executadas para entregar um parte do produto gerado no método. Enquanto que as tarefas se referem a

divisão das atividades em pequenas ações ou conjunto de passos a serem desenvolvidos para efetuar a intenção proposta em cada etapa.

De acordo com essa categorização, foram definidas três etapas para a utilização do *Agile ETeasy*: **Planejamento**, **Execução** e **Encerramento**, conforme apresentado na Figura 4.3. Em cada etapa, para a condução das atividades foram definidas um conjunto de tarefas associadas ao uso de técnicas específicas, para auxiliar na obtenção dos resultados esperados. Inicialmente, as etapas foram planejadas para serem executadas da seguinte forma:

1. **Etapa de Planejamento:** Nesta etapa é sugerido que seja *criada uma hipótese a ser testada*. Para isso, são indicadas a realização das tarefas: conhecer o produto e responder à questão “Qual o teste mais importante que eu posso realizar agora?”. Por fim, as técnicas de *Brainwriting* e pesquisa documental também são recomendadas.
2. **Etapa de Execução:** Nesta etapa deve-se *aplicar o Teste Exploratório e observar os resultados*. Para isso, são indicadas a realização das tarefas: iniciar a exploração do sistema e analisar o comportamento do *software*, considerando os requisitos definidos. Por fim, as técnicas de *Checklist* de Heurística e *Pair review* também são recomendadas.
3. **Etapa de Encerramento:** Nesta etapa deve-se *registrar o teste executado*. E, *registrar e alinhar com a equipe as melhorias a serem implementadas ou corrigidas*. Para isso, são indicadas a realização das tarefas: descrever as informações encontradas sobre a hipótese testada, apresentar as percepções sobre a hipótese testada, destacar os *bugs* encontrados e apontar sugestões de melhorias a respeito dos *bugs* encontrados. Por fim, as técnicas de *Workshop* com os membros da equipe ou *Brainstorming* com *stakeholders* também são recomendadas.

Com o uso do método *Agile ETeasy* são gerados dois artefatos de teste: um artefato de planejamento do ET, na etapa de Planejamento; e, um relatório do ET, na etapa de Encerramento - conforme apresentado na Figura 4.4.

É importante destacar que os elementos aqui citados constituem a primeira versão do método *Agile ETeasy*, e que foram refinados sucessivas vezes, durante a DSR, até chegar em sua versão final. Os elementos que compõem cada etapa do método *Agile ETeasy* são detalhados, em sua versão final, no Capítulo 3.

| Data: | | | | <dd/mm/aaaa> | |
|-------------|---|-------------------------------------|--|--|--|
| Sistema: | | | | <identificação do nome do sistema/projeto> | |
| Versão: | | | | X.X.X | |
| Sprint: | | | | <caracterização da Sprint (nome, numeração ou referência)> | |
| Testadores: | | | | <identificação do profissional que realizará (ão) o teste> | |
| ID | Referência ao Requisito | Escopo a ser testado | Testadores | | |
| 01 | <referência à Use Case, User Stories, documento, ou Sistema de Gerenciamento de Requisitos> | <descrição do escopo a ser testado> | <identificação do profissional (s) responsável (s) por testar o referido escopo> | | |
| 02 | | | | | |
| 03 | | | | | |
| 04 | | | | | |
| 06 | | | | | |
| 07 | | | | | |
| 08 | | | | | |
| 09 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |

| Relatório | |
|------------------|--|
| Testadores | <identificação do profissional (s) responsável (s) por testar o referido escopo> |
| Esopo testado | <descrição do escopo a ser testado> |
| Arquivo de dados | Usuários: <perfil/descrição do tipo de usuário> Configuração: <configuração do sistema, em que o teste foi executado> Data/Numero/String: <data/numero/string que referencia a funcionalidade testada no sistema> Arquivos: <arquivos necessários para execução da funcionalidade de teste> Navegação: <forma de navegação no sistema> Outros: <outras informações importantes a serem registradas> |
| Defeitos | 1. <descrição dos defeitos encontrados> 2. 3. |
| Melhorias | 1. <sugestões de melhorias a serem implementadas a partir dos defeitos percebidos> 2. 3. |
| Pendências | 1. <funcionalidades que não foram finalizadas, ou seja, pendências percebidas durante o teste> 2. 3. |
| Notas | 1. <notas adicionais sobre o teste executado> 2. 3. |
| Links | 1. <links de acesso à registros do teste, como gifs ou prints de tela dos bugs encontrados> 2. 3. |
| Anexos | 1. <documentos adicionais inseridos como anexo, necessários para a execução ou compreensão da funcionalidade testada> 2. 3. |

Figura 4.4: Primeira versão dos artefatos de teste do método *Agile ETeasy*.

Por fim, nesta etapa de *Design da Solução* também foi criado um *website* do método *Agile ETeasy*⁷. Inicialmente, o *website* foi disponibilizado para a validação do método na etapa de *Validação da Solução* dos Ciclos 1 e 2, desta DSR. No geral, é possível visualizar o detalhamento das etapas, atividades, tarefas, técnicas e elementos de saída contidos no método *Agile ETeasy*. Também é possível visualizar um vídeo auto-explicativo e imagens que ilustram o método. A Figura 4.5 apresenta a página inicial do *website* do método *Agile ETeasy*, contendo uma apresentação geral do método.

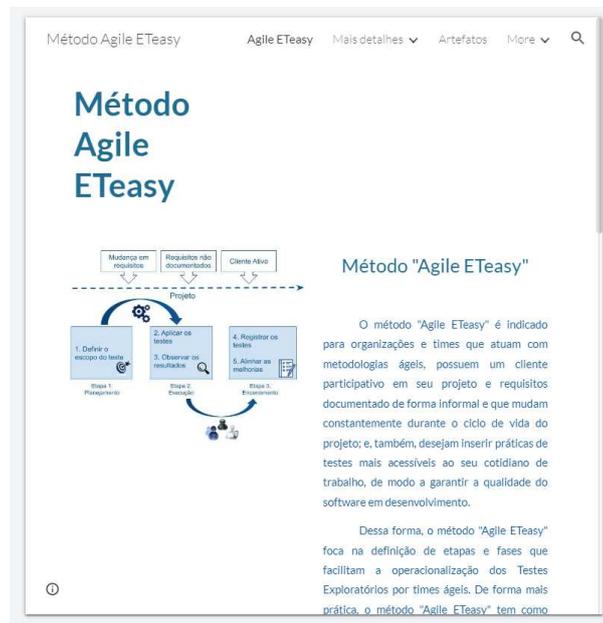


Figura 4.5: *Print Screen* do *website* do método *Agile ETeasy*.

⁷Acesso ao *Website* do método *Agile ETeasy* <https://bitly.com/BIJTEpQg>

4.2.3 Validação da Solução

Com a intenção de validar o método *Agile ETeasy*, foi realizada uma validação empírica com profissionais de equipes ágeis, denominados nesta pesquisa como “avaliadores”. Assim, foi conduzida uma pesquisa de opinião, onde os avaliadores deveriam fornecer sua opinião sobre o entendimento e a utilidade do método *Agile ETeasy*, bem como sugestões de melhoria e críticas ao método. Os resultados foram sintetizados em um relatório de dados⁸.

Inicialmente, com a finalidade de caracterizar o perfil profissional dos avaliadores, foram coletadas informações sobre sua instrução acadêmica e sua experiência profissional na área de Teste de *Software*. É importante ressaltar que a identificação dos avaliadores não foi apresentada, nesta pesquisa, para garantir o anonimato dos participantes.

Em relação ao nível de formação acadêmica dos participantes, 40% possuem uma graduação completa, enquanto que 40% possuem uma pós-graduação *stricto sensu* em nível de mestrado, e apenas 20% ainda está cursando um doutorado. Outro fator observado foi a experiência profissional dos participantes em relação à execução de práticas de Teste de *Software* no contexto ágil. Foi percebido que 20% dos avaliadores possuem mais de 5 anos de experiência com ST ágil, enquanto que 20% deles possuem entre 3 e 4 anos de experiência e, outros 20% possuem entre 1 e 2 anos de experiência. É importante ressaltar que 40% dos avaliadores possuem menos de 1 ano de experiência. Esse aspecto por ser justificado porque, no contexto ágil, os times são multifuncionais, ou seja, um membro pode executar diferentes papéis de acordo com a necessidade do projeto. Isso pode não ter proporcionado ao avaliador tempo de experiência com ST, especificamente.

Sobre a experiência com ST no contexto ágil, apenas um avaliador informou ter pouca experiência com Teste de *Software* e ET, mais especificamente. Isso justifica o pouco *feedback* recebido por este avaliador, em questões subjetivas no questionário de validação do método *Agile ETeasy*.

No geral, sobre a facilidade de entendimento do método, 80% dos avaliadores indicaram que foi fácil entender as etapas, atividades e tarefas do método; bem como, compreender com clareza como utilizar as técnicas (*Brainwriting*, Pesquisa documental, *Pair review*, *Workshop* e *Braisntorming*) sugeridas nas atividades (60% dos avaliadores) e preencher os artefatos,

⁸Síntese dos dados do Questionário da etapa de Validação da Solução do Ciclo 1: <https://bitly.com/JPzpjczC>

de planejamento e o relatório de encerramento (60% dos avaliadores). Em síntese, esses resultados indicam que os elementos (etapas, atividades, tarefas, técnicas e artefatos) que compõem o método *Agile ETeasy* são fáceis de entender, embora os avaliadores considerem que as tarefas necessitem de uma descrição mais detalhada e clara. De modo complementar, para facilitar o entendimento do artefato de planejamento, foi fornecida a seguinte sugestão de melhoria: “*Talvez adicionar um campo arquivo (opcional) no artefato de planejamento. Às vezes pode ser necessário especificar um arquivo de configuração necessário para testar um requisito*” (Avaliador 5).

Sobre a utilidade do método *Agile ETeasy*, os avaliadores concordam, em sua maioria, que as etapas, atividades e tarefas descritas no método são adequadas para a realização de Testes Exploratórios. Também foi constatado, por 80% dos avaliadores, que as técnicas de *Brainwriting* e de Pesquisa Documental são úteis à aplicação da etapa de Planejamento. Em sua maioria, aproximadamente 80% dos avaliadores concordaram que a técnica de *Checklist* de Heurísticas é útil à etapa de Execução. Enquanto as técnicas de *Workshop* e *Brainstorming*, também foram consideradas úteis à etapa de Encerramento, por 80% dos avaliadores. Entretanto, é importante destacar que um dos avaliadores não considera útil a técnica de Pesquisa Documental na etapa de Planejamento, enquanto que outro avaliador também considera pouco útil a técnica de *Workshop*, na etapa de Encerramento. Em síntese, esses resultados indicam, em sua maioria, que os elementos (etapas, atividades, tarefas, técnicas e artefatos) que compõem o método *Agile ETeasy* são úteis para executar Testes Exploratórios.

Sobre a experiência em ler a descrição do método, alguns avaliadores afirmaram que:

- Avaliador 2: “*O método é fácil de entender e parece simples de aplicar, sem precisar de muita experiência com Testes Exploratórios.*”
- Avaliador 3: “*Boa, a explicação das etapas estão bem definidas, mas sinto falta de exemplos reais. Acho que ficaria mais claro ainda principalmente para pessoas com poucas experiências.*”
- Avaliador 4: “*O texto sobre o método está bem escrito, no entanto senti falta de definições iniciais mais claras sobre Teste Exploratório, sua aplicabilidade, vantagens, desvantagens, diferenças com testes Ad-Hoc, etc.*”

Os avaliadores também indicaram algumas críticas e sugestões de melhorias a serem consideradas no refinamento do método *Agile ETeasy*. É importante destacar que não houveram sugestões de edição (remoção, alteração ou adição) nas atividades, nas tarefas ou nas técnicas indicadas no método *Agile ETeasy*. A seguir apresentamos algumas principais sugestões fornecidas.

- Avaliador 3: “Acredito que o método tem os pontos necessários para uma análise de testes exploratório.”
- Avaliador 5: “Talvez adicionar uma opção de arquivo para cada escopo do planejamento. As vezes é necessário especificar um arquivo de configuração específico. Além disso, definir vários escopos para um único requisito (embora isso possa ser feito durante o preenchimento)”.

Em síntese, ao final da etapa de *Validação da Solução* as sugestões de melhorias e críticas foram refletidas e aplicadas ao método *Agile ETeasy*. Assim, foram aplicadas as seguintes alterações: (i) foi inserido no *website*, um exemplo real da aplicação do método *Agile ETeasy* em um sistema *web* (conforme sugerido pelo Avaliador 3); (ii) foi alterada a descrição do campo ‘Referência ao Requisito’ do artefato de planejamento do teste, para esclarecer a informação que deve ser inserida (conforme sugerido pelo Avaliador 4); (iii) as técnicas de Pesquisa Documental e *Workshop* foram reescritas para esclarecer seu entendimento, e confirmar a avaliação de sua utilidade na etapa de *Implementação da Solução*, em um contexto real. A sugestão do Avaliador 4 sobre adicionar conceitos de ET e comparações com outros tipos de teste não foram aplicadas ao método no *website* porque não compreendiam o escopo desta proposta (descrição do método *Agile ETeasy*).

4.2.4 Implementação da Solução

No intuito de coletar sugestões de melhorias ao artefato proposto nesta tese, foi aplicada a etapa de *Implementação da Solução* por meio de um Estudo de Caso 1 com uma equipe ágil composta por três profissionais (desenvolvedores). Assim, o método *Agile ETeasy* foi implementado em um projeto de desenvolvimento de sistemas embarcados. É importante destacar que a etapa de *Implementação da Solução* foi planejada e conduzida após o *feedback* obtido por meio dos avaliadores na etapa de *Validação da Solução*.

Inicialmente, foram coletadas algumas informações para caracterização dos participantes deste estudo. É importante destacar que os participantes não foram identificados, nesta tese, para garantir o anonimato das informações. Desse modo, foram extraídas informações sobre a experiência profissional dos participantes com práticas de Teste de *Software*, no contexto ágil.

Cada um dos participantes deste estudo possui um nível de formação acadêmica distinto. Um deles, possui uma graduação completa; outro, uma graduação em andamento; e, outro, uma pós-graduação *stricto sensu* em nível de mestrado. Sobre a prática de Teste de *Software*, no contexto ágil, dois participantes informaram ter entre 2 e 3 anos de experiência com a execução de testes em projetos de desenvolvimento de *software*; enquanto que outro participante informou possuir menos de 1 ano de experiência com testes. É possível que o participante menos experiente não tenha tido a oportunidade de atuar com testes em outros projetos porque se trata de um participante em finalização de estudos na graduação, ou seja, recém-ingresso no mercado de trabalho. Ainda sobre a experiência com práticas de teste, no contexto ágil, os participantes informaram que:

- Participante 1: *“Tenho pouca experiência com a área de teste de software. Minha experiência nesta área vem do projeto que participo. Geralmente, faço testes com minhas implementações e quando estou revisando implementações de companheiros de equipe.”*
- Participante 2: *“Participei, durante a graduação, de um projeto exclusivamente como testador. Nesse projeto, utilizei ferramentas de análise estática, e.g. TestLink. Além dessa experiência, cursei algumas disciplinas na graduação sobre testes e qualidade de software.”*
- Participante 3: *“Uso técnicas de teste para ajudar a equipe nas várias etapas de verificação do software.”*

Todos os participantes informaram que possuem o hábito de testar todas as funcionalidades que desenvolvem. A seguir apresentamos as informações fornecidas pelos participantes sobre como desenvolvem os testes, considerando tipo de teste desenvolvido, periodicidade, dentre outros aspectos.

- Participante 1: *“Normalmente, faço testes de unidade e regressão sempre que estou implementando algo novo. Realizo os testes sempre antes de enviar uma nova implementação.”*
- Participante 2: *“Geralmente os testes são feitos a cada módulo/funcionalidade implementado no sistema durante o próprio desenvolvimento e ao final do mesmo. Os testes são feitos de maneira randômica, seguindo apenas o "feeling" do que precisa ser testado de acordo com a funcionalidade e o que foi modificado no código.”*
- Participante 3: *“Semanal e individualmente, uso técnicas de teste funcional (caixa-preta), teste estrutural (caixa-branca), teste de regressão e teste de usabilidade.”*

A seguir são apresentadas as informações fornecidas pelos participantes sobre sua experiência com Testes Exploratórios. De forma complementar, foi constatado que nenhum dos participantes chegou a participar de projetos que adotam Testes Exploratórios como prática de teste.

- Participante 1: *“Não tenho experiência com testes exploratórios.”*
- Participante 2: *“A única experiência que tive com esse tipo de testes foi durante um curso ministrado no ano de 2021 dentro do projeto que atualmente participo.”*
- Participante 3: *“Às vezes, uso técnicas de Teste Exploratório, principalmente quando não tenho fácil acesso a documentos de requisitos do software em tempo de teste.”*

Após a coleta de informações sobre a caracterização dos participantes, um mini-curso⁹ sobre o método *Agile ETeasy* foi conduzido com os participantes. Este mini-curso teve a duração de, aproximadamente, 40 minutos. Além de apresentar o método, foi feita uma contextualização sobre ET. Em seguida, os participantes foram conduzidos a implementar o método *Agile ETeasy* em um módulo desenvolvido no projeto em que atuam.

Em síntese, foram testados sete escopos de teste referentes à sete funcionalidades do projeto. Como o sistema testado se referiu a um sistema embarcado (terminal eletrônico de maquineta de cartão) e todos estavam aplicando os testes remotamente, um desenvolvedor executou o teste e narrou a execução, e, os outros 2 acompanharam, atentamente, a execução

⁹Acesso ao material adotado no mini-curso: <https://bitly.com/dRwAoiJy>.

do teste, auxiliando com o registro do tempo de duração do ET, com a verificação e organização dos arquivos usados, com sugestões do que verificar no ET, dentre outras atividades.

Dentre os testes realizados, apenas uma funcionalidade precisou de anotações sobre melhorias a serem implementadas. As demais funcionalidades não apresentaram falhas ou inconsistências. No geral, testar essas funcionalidades por meio do método *Agile ETeasy* durou, em média, 2 horas e 30 minutos - considerando todas as etapas sugeridas: planejamento, execução e encerramento.

É válido destacar que na Reunião de Alinhamento, na etapa de Encerramento, surgiu uma sugestão de melhoria que não havia sido percebida durante o Relatório de Encerramento. Durante a aplicação do método, um participante executou e narrou o teste; enquanto que os outros dois participantes acompanhavam atentamente a execução do teste, e realizavam tarefas como verificar arquivos a serem usados, cronometrar o tempo gasto, fornecer sugestões do que verificar, dentre outros.

A discussão da avaliação aplicada nesta etapa de *Implementação da Solução* é ampliada na Seção 4.3.1, onde são apresentados os resultados da etapa de *Avaliação da Implementação*, iniciada no Ciclo 2.

4.3 Ciclo 2 da DSR

Neste capítulo são apresentados os resultados das etapas de *Avaliação da Implementação*, *Design da Solução*, *Validação da Solução* e *Avaliação da Implementação* do Ciclo 2 da *Design Science Research*, de acordo com os procedimentos metodológicos descritos na Seção 4.1.2.

4.3.1 Avaliação da Implementação

Com o intuito de avaliar a implementação do método *Agile ETeasy* sobre a facilidade de entendimento e utilidade dos elementos e dos artefatos, após a execução da etapa de *Implementação da Solução* do Ciclo 1, foram aplicados um Diário de Bordo *online* e entrevistas com os participantes do Estudo de Caso 1. A avaliação conduzida nesta etapa foi importante para mais um refinamento no método.

Cada participante registrou, no seu Diário de Bordo (consultar *template* no Apêndice

I), ao final da implementação do método, sua experiência na utilização dos elementos do método *Agile ETeasy*, os pontos positivos, os pontos negativos, os resultados alcançados e/ou as dificuldades encontradas. As informações registradas no Diário de Bordo foram sintetizadas e também embasaram mais um refinamento do método *Agile ETeasy*, ao final do Estudo de Caso 1. Alguns comentários principais registrados no Diário de Bordo, como a identificação do participante, o elemento do método *Agile ETeasy*, ao qual o comentário se refere; e, o comentário registrado pelo participante, são apresentados na Tabela 4.7.

Os participantes também foram entrevistados de forma coletiva, pois ambos possuíam um perfil comum e aplicaram o método *Agile ETeasy* em conjunto durante o Estudo de Caso 1. A síntese das entrevistas pode ser verificada em <https://bitly.com/YyUofjXu>.

Inicialmente, os participantes foram questionados sobre a facilidade de entendimento do método *Agile ETeasy*, considerando as etapas, atividades e tarefas compõem. A seguir, é possível observar a opinião dos três participantes, do Estudo de Caso 1:

- Participante 1: *“Eu também achei bem simples, bem, bem objetivo. Não acredito que uma pessoa vai ter muita dificuldade de aplicar.”*
- Participante 2: *“Sim. É, consegui sim compreender com clareza.”*
- Participante 3: *“Sim, numa escala de zero a 10. Acho que 9. Eu só senti um pouco de dificuldade no relatório de encerramento. Sim, só isso mesmo!”*

Não houveram sugestões adicionais para facilitar o entendimento das etapas, atividades e tarefas. Entretanto, em relação às técnicas indicadas no método *Agile ETeasy*, um dos participantes destacou que a técnica *Checklist* de Heurísticas não foi aplicada. Enquanto que, as demais técnicas foram usadas sem que os participantes percebessem que as estavam adotando.

Sobre as tarefas indicadas no método *Agile ETeasy*, os participantes relataram que não sentiram dificuldades em compreendê-las, e afirmaram o seguinte: *“Bem objetivo. Fácil de entender o que tem que se feito.”* e *“Eu achei bem objetivo e resumido. Eu gostei bastante do que têm nos slides. O método tá bem prático.”*

Sobre os artefatos de teste gerados nas etapas de Planejamento e Encerramento do método *Agile ETeasy*, os participantes indicaram a necessidade de melhorar a descrição de alguns campos do método, de modo a deixá-los mais claros. Assim, foram sugeridas alterações

Tabela 4.7: Síntese dos registros no Diário de Bordo.

| Participante | Elemento | Registro no Diário de Bordo |
|----------------|-------------------------------------|---|
| Participante 1 | Relatório de Encerramento | <p>Comentário 1: “<i>Os campos Anexos e Links geraram um pouco de ambiguidade na minha opinião. Dependendo do escopo do teste a ser realizado, ficava muito difícil definir em qual campo inserir a informação.</i>”</p> <p>Comentário 2: “<i>Melhorar o campo ‘Data/Número/String’. Não ficou muito claro o que o campo pede.</i>”</p> <p>Comentário 3: “<i>Adicionar um campo para descrever os equipamentos utilizados para realizar o teste.</i>”</p> |
| Participante 2 | - | Não informou nenhum comentário ou observação no Diário de Bordo. |
| Participante 3 | Relatório de Encerramento | Comentário 1: “ <i>Os termos ‘Links’, ‘Anexos’, ‘Usuários’, ‘Configuração’, ‘Data/número/string’, ‘Arquivos’, ‘Navegação’ e ‘outros’ parecem ser vagos e é difícil decidir o que se espera ser informado.</i> ” |
| | Atividade Definir o escopo de teste | Comentário 1: “ <i>Recomendar exemplos de escopos de teste válidos e viáveis.</i> ” |

nos campos do artefato de Encerramento: ‘Links’, ‘Anexos’, ‘Usuários’, ‘Configuração’, ‘Data/número/string’, ‘Arquivos’, ‘Navegação’ e ‘outros’.

Em seguida, os participantes foram questionados sobre a utilidade das etapas, atividades e tarefas do método *Agile ETeasy*. A seguir apresentamos as principais observações dos participantes:

- Participante 1: *“Eu achei que se complementam. Então, não removeria, mas também não adicionaria mais etapas.”*
- Participante 2: *“É eu achei que ficou bem. Você não consegue fazer execução, se não fizer planejamento e você não consegue encerrar execução de forma correta se não seguir as etapas que estão descritas. Também não adicionaria mais etapas, também não removeria nenhuma das três.”*
- Participante 3: *“É, eu achei que foram úteis sim.”*

De modo complementar, o Participante 1 sugeriu adicionar uma seleção de profissionais para executar o ET, logo na fase de planejamento: *“Eu acho que na parte de planejamento talvez tenha uma etapa para selecionar equipe. Sim, de quem conhece bem um produto... escolher pessoas que conhecem o produto, mas não participa do desenvolvimento e conhecem participaram do desenvolvimento. Ter uma seleção de uma equipe que não fica enviada.”*. Entretanto, é viável realizar uma alocação dos membros do time para aplicar o ET, considerando a funcionalidade que será testada e o responsável por seu desenvolvimento.

Sobre a utilidade das técnicas indicadas no método, ambos os participantes informaram que gostariam de continuar usando o método por mais tempo para poder confirmar sua utilidade. Ou seja, o tempo utilizado no Estudo de Caso não foi suficiente para obter essa confirmação e não foi possível prosseguir o acompanhamento da implementação do método com este time, pois no período de realização desta pesquisa, não houveram outras funcionalidades recém-implementadas pelo time, para que o ET pudesse ser aplicado.

Os participantes também informaram a necessidade de ter acesso à um guia com orientações sobre o método, por meio das seguintes afirmações: *“Como foi a nossa primeira experiência, eu acho que a explicação é necessária. Eu acho que ainda mais para quem está tendo um primeiro contato com o método.”* (Participante 1); e, *“Uma explicação é necessária e é bom ter um guia também.”* (Participante 3).

Também foi apontada a necessidade de automatização do método por meio de uma ferramenta que facilitasse a geração dos artefatos de testes, ao informarem o seguinte: “*Uma ferramenta seria essencial para melhorar o método, ajudaria muito.*” (participante 1) e “*Eu acho que a experiência com o método seria muito mais proveitosa se houvesse alguma ferramenta para dar suporte ao que fizemos nas planilhas. Por exemplo, anexar links e conectar coisas*” (Participante 2).

Algumas outras observações adicionais foram fornecidas pelos participantes. Essas informações são apresentadas a seguir:

- Participante 1: “*Mas eu gostei, eu gostei. Assim, como primeira experiência, eu gostei. Eu usaria no meu projeto. Acho que ele tem potencial para ajudar no teste da aplicação. E, outra coisa, que o participante 3 falou também: como tem muita gente trabalhando remoto, isso iria melhorar muito a comunicação, porque a pessoa fez os testes em casa, e com os artefatos bem documentados, e alguém que não tá trabalhando junto esse tempo todo, consegue acompanhar melhor o que tá sendo realizado no trabalho.*”
- Participante 3: “*Estou com a impressão aqui que muito do que a gente viu hoje em forma de método, a gente já faz, mas sem ser seguir um método... Mas, acho que vá ajudar nessa na captura dessas informações. Por exemplo: às vezes eu faço uns testes, aí eu vejo como pode melhorar em alguma funcionalidade, em um certo aspecto, mas eu só vou falar sobre isso depois, quando eu estou lá pessoalmente, porque é quando eu sinto mais facilidade para me expressar... Talvez esses artefatos ajudem nisso. E também as pessoas que estão trabalhando remoto, que estão distante, que às vezes fazem testes e não se comunicam. Bem, talvez seja útil nesse sentido.*”

4.3.2 Design da Solução

O *feedback* dos participantes do Estudo de Caso foram considerados para mais um refinamento do artefato gerado nesta DSR, o método *Agile ETeasy*. Desse modo, para o desenvolvimento desta etapa de *Design da Solução* foram consideradas as informações extraídas do Diário de Bordo e das entrevistas aplicadas com os participantes - conduzidas nas etapas de *Implementação da Solução* do Ciclo 1, e *Avaliação da Implementação*, do Ciclo 2.

Assim, constatamos a necessidade de melhorias nos artefatos, relacionadas à edição, remoção, criação ou detalhamento da descrição de alguns campos de dados. No Relatório de Encerramento, por exemplo, os campos “links”, “data/número/string”, “navegação” e “equipamento” foram removidos porque não se adequavam ao contexto de aplicação do método; já os campos “anexos”, “outros”, “usuário”, “configuração” e “arquivos”, tiveram sua descrição reescritas para uma melhor compreensão do seu objetivo. Nenhum novo campo foi adicionado aos artefatos.

Outra sugestão fornecida pelos participantes foi a inserção de um guia com orientações mais precisas sobre o método *Agile ETeasy*. Desse modo, foi desenvolvido um vídeo explicativo, de curta duração, detalhando os principais elementos do método. Este vídeo foi inserido no *website*, de modo a contribuir com as próximas etapas de validação do artefato.

Algumas sugestões de orientação para aplicação do método também foram fornecidas pelos participantes, como: (i) “Um desenvolvedor que não implementou a funcionalidade é quem deve testar, para evitar viés no teste.” (Participante 2); (ii) “A explicação do método é necessária e é bom ter um guia também - para uma primeira experiência com o método.” (Participante 3); e, (iii) “Recomendar exemplos de escopos de teste válidos e viáveis.” (Participante 2). Outras considerações foram feitas, mas já estavam implementadas nas orientações de aplicação do método. Um outro aspecto observado foi que a técnica *Checklist* de Heurísticas não foi utilizada pelos participantes no Estudo de Caso 1.

Em síntese, todas as recomendações de edição nos artefatos foram implementadas. Foram poucas as sugestões de orientações para aplicação prática do método *Agile ETeasy* e de considerações sobre mudanças nos demais elementos do método, como etapas, atividades ou técnicas. Por esse motivo, não houveram mudanças impactantes na proposta do “*Agile ETeasy*”.

Portanto, o refinamento aplicado no método *Agile ETeasy*, nesta etapa de *Design da Solução* do Ciclo 2, contribuiu com a implementação de melhorias aos elementos que o compõem - principalmente, no artefato de Relatório de Encerramento - e gerou uma nova versão do método para ser validada na próxima etapa da DSR, a etapa de *Validação da Solução*, descrita a seguir. A versão final do método *Agile ETeasy* é apresentada, em detalhes, no Capítulo 3.

4.3.3 Validação da Solução

A etapa de *Validação de Solução* foi aplicada pela segunda vez nesta DSR, com o intuito de validar, novamente, o método *Agile ETeasy* após os refinamentos aplicados nas etapas anteriores. Para isso, foi realizada a segunda validação empírica com profissionais de equipes ágeis, também denominados nesta pesquisa como “avaliadores”.

De modo complementar, o método *Agile ETeasy* também foi aplicado em uma turma de alunos de uma disciplina de Teste de *Software* de um curso de graduação em Ciência da Computação da Universidade Federal de Campina Grande. Desse modo, foi coletado o *feedback* de alguns alunos que implementaram o método em seus projetos acadêmicos e o *feedback* da docente responsável pela disciplina. Como a participação na validação, por meio do preenchimento do questionário *online*, foi opcional, apenas três alunos forneceram sua opinião sobre o método. É importante destacar que esses alunos já possuem experiência profissional na área, visto que são alunos concluintes do curso de graduação. Por este motivo, suas respostas foram mantidas na pesquisa.

Do mesmo modo que no Ciclo 1 desta DSR, também foi conduzida uma pesquisa de opinião, onde os avaliadores deveriam fornecer sua opinião sobre o entendimento e a utilidade do método *Agile ETeasy*, bem como sugestões de melhoria e críticas. As informações obtidas foram sintetizadas em um relatório de dados¹⁰.

Primeiramente, com a finalidade de caracterizar o perfil profissional dos avaliadores, foram coletadas informações sobre sua instrução acadêmica e sua experiência profissional na área de Teste de *Software*. Assim sendo, a identificação dos avaliadores não foi apresentada, nesta pesquisa, para garantir o anonimato dos participantes.

Em relação ao nível de formação acadêmica dos participantes, 41.7% possuem uma graduação completa, 25% possuem uma graduação em andamento, enquanto que 16.7% possuem uma pós-graduação *stricto sensu* em nível de mestrado também em andamento, e 25% concluiu uma pós-graduação *stricto sensu* em nível de mestrado.

Outro fator observado foi a experiência profissional dos participantes em relação à execução de práticas de *Agile ST*. Foi percebido que 8.3% dos avaliadores possuem entre 4 e 5 anos de experiência com *Agile ST*, enquanto que 25% deles, possuem entre 3 e 4 anos de

¹⁰Síntese dos dados do Questionário da etapa de Validação da Solução do Ciclo 2: <https://bitly.com/KkxAjuJw>

experiência e, outros 41.7% possuem entre 2 e 3 anos de experiência. É importante ressaltar que, 16.7% dos avaliadores possuem entre 1 e 2 anos de experiência com *Agile ST*, e apenas 8.3% afirmaram ter menos de 1 ano de experiência. O pouco tempo de experiência de alguns avaliadores (menos de 2 anos) pode ser justificado porque alguns destes profissionais ainda estão em início de carreira profissional, como os avaliadores que ainda cursam uma graduação enquanto atuam em projetos de desenvolvimento de *software*.

Sobre a experiência com *Agile ST*, apenas um avaliador não informou desenvolver testes nos projetos em que atua/atuou. Os demais avaliadores relataram aplicar testes após a implementação de funcionalidades no projeto. Foi percebido que os avaliadores, em sua maioria, costumam aplicar testes às funcionalidades que desenvolvem. Dentre as práticas de teste realizadas, percebemos que as mais comuns são testes de unidade e testes de integração, testes manuais, regressivos, teste de componentes e testes exploratórios. Alguns *frameworks* também tem sido adotados, como o *framework Jasmine* e *Karma*, e a Biblioteca *pytest*.

Sobre a experiência com Testes Exploratórios, 50% dos avaliadores informaram ter executado ET em projetos que atuaram. Apenas 8.3% deles informou ter participado de um projeto acadêmico. Enquanto que 41.6% deles não forneceram resposta para este questionamento. As respostas fornecidas pelos avaliadores são apresentadas a seguir:

- *“Geralmente utilizava o teste exploratório logo após os testes de script manuais (onde eram testados as especificações do documento de requisitos). Ele era usado como coadjuvante para testar as funcionalidades do sistema buscando erros ou bugs não previstos nas especificações do sistema, mas que pudessem trazer algum prejuízo à aplicação.”*
- *“No projeto que estou trabalhando atualmente os testes exploratórios são aplicados sempre que ocorre um deploy contendo um pacote de novas funcionalidades.”*
- *“Sim, a maioria dos testes que construí foram exploratórios, onde o testador estuda a aplicação, elenca os possíveis casos de teste e só então implementa o teste.”*
- *“Sim. Já participei, mas de forma muito informal.”*
- *“Acho que já apliquei, mas de forma mais simples... não segui uma estratégia específica e também não conheço estratégias de teste exploratório.”*

- *“Sempre aplico Teste Exploratório. Principalmente, quando não temos documentos do projeto a seguir. Mas, não aplico nenhum processo específico. É mais exploração do software mesmo. Listo as funcionalidades que preciso testar, e vou explorando possíveis caminhos nelas. É assim que faço, geralmente. Anoto os bugs e depois converso com a equipe o que precisa ser mudado.”*

Em síntese, sobre a facilidade de entendimento do método *Agile ETeasy* foi possível constatar uma predominância nas respostas no seguimento de concordância, no que se refere ao entendimento das etapas, atividades e tarefas, bem como das técnicas (*Brainwriting*, Pesquisa documental, *Pair review*, *Workshop*, *Braisntorming*) indicadas para utilização do método - o que pode simbolizar que os avaliadores consideram fáceis de entender os elementos que compõem o método *Agile ETeasy*. Para mais detalhes consultar o relatório de dados deste estudo, disponível em: <https://bitly.com/KkxAjuJw>.

Em relação aos artefatos de teste, 83.3% dos avaliadores concordou que o artefato de planejamento do teste e o relatório de encerramento são de fácil entendimento. Entretanto, 16.4% deles discorda que os artefatos estão compreensíveis. Em síntese, esses resultados indicam que os elementos (etapas, atividades, tarefas, técnicas e artefatos) que compõem o método *Agile ETeasy* são fáceis de entender, embora alguns poucos avaliadores considerem que os artefatos necessitam ser esclarecidos para um melhor entendimento.

Por fim, 25% dos avaliadores forneceram algumas sugestões de melhoria relacionadas às etapas, atividades e tarefas:

- *“Com relação à primeira atividade da parte da execução, ‘Participam desta etapa os profissionais envolvidos no desenvolvimento da aplicação e que executarão os testes - um desenvolvedor e o líder do time’, senti falta da figura do testador, pois, na maioria dos projetos em que trabalhei quem executava essa parte de testes eram profissionais que não eram responsáveis pelo desenvolvimento. Que entram na sprint após o desenvolvimento liberar as funcionalidades para teste. Fiquei confusa em distinguir em que momento da sprint a execução acontecerá”*
- *“Descrever de forma clara as informações que se quer ajuda bastante. Ter critérios de aceite e definições.”*
- *“Acho que disponibilizar um passo a passo simples e direto ajudaria muito.”*

Para facilitar o entendimento dos artefatos gerados no método *Agile ETeasy*, algumas sugestões de melhorias também foram fornecidas pelos avaliadores, a citar: “Disponibilizar um exemplo dos artefatos preenchidos e com contexto explicativo ajudaria bastante” e “Poderia ser algo mais automático. Uma ferramenta, talvez ajudasse no preenchimento.”.

Em relação à utilidade do método *Agile ETeasy* foi possível constatar uma predominância nas respostas no seguimento da concordância, para as etapas, atividades e tarefas definidas. O mesmo foi constatado para as técnicas de *Brainwriting* e Pesquisa Documental, indicadas para a fase de Planejamento. Entretanto, 8.3% dos avaliadores não concordou que a técnica de Pesquisa Documental é útil na etapa de Planejamento.

A técnica de *Checklist* de Heurísticas, na etapa de Execução, também foi avaliada como útil à etapa de Execução, de acordo com as respostas dos avaliadores (66.6%, no total). Todavia, foi constatado uma discordância de 16.6% dos avaliadores, enquanto que outros 16.6% não conseguem opinar sobre a concordância ou discordância neste aspecto.

Em sua maioria, equivalente a 75% dos avaliadores, as técnicas de *Workshop* e *Brainstorming* também foram consideradas úteis na etapa de Encerramento. Foi percebido também que 25% dos avaliadores não conseguem opinar sobre a concordância ou discordância neste aspecto.

Em síntese, esses resultados indicam, em sua maioria, que os elementos (etapas, atividades, tarefas, técnicas e artefatos) que compõem o método *Agile ETeasy* são úteis para executar Testes Exploratórios. Embora, alguns poucos elementos ainda precisem de melhorias para serem aperfeiçoados.

De forma complementar, os avaliadores forneceram um *feedback* avaliativo adicional sobre o método *Agile ETeasy*, por meio de críticas, sugestões de melhorias, inclusão ou remoção de algum elemento do método, como pode ser acompanhado a seguir:

- “Não remover o *BRAINSTORMING*, mas utilizar dependendo do contexto. Acredito que o uso do mesmo, vai depender do tamanho das novas funcionalidades, pois se for algo ‘pequeno’, acredito que o *WORKSHOP* se encaixaria melhor.”
- “Remover *TÉCNICA DE WORKSHOP*, não entendo sua finalidade.”
- “Talvez repensar sobre simplificar essa lista de heurísticas.”

Sobre a experiência em ler a descrição do método no *website*, alguns avaliadores afirmaram que: *“Bem agradável as informações estão bem explicadas.”*, *“O vídeo ajudou a compreender com mais facilidade o método.”*, *“O site está fácil de entender e o método também.”*, dentre outros comentários. Sobre sugestões de melhorias para o método *Agile ETeasy*, alguns avaliadores sugeriram o seguinte:

- *“Também seria importante dizer em que etapa do desenvolvimento o teste exploratório pode/deve ocorrer? Se em todas as fases até a funcionalidade entrar em produção, como eu penso, assim como o testador, o product owner teria alguma responsabilidade no método, uma vez que sua expertise tem muito à acrescentar ao projeto? Gostaria de deixar essa reflexão.”*
- *“Checklist de Heurísticas é interessante. Ainda não tinha pensando nisso. Mas, esse documento de checklist que é apresentado, é bem longo pra lermos e pensarmos no que extrair e aplicar depois. Acho que poderia ter uma orientação mais simples dessas heurísticas.”*
- *“Acho que a lista de heurísticas poderia ser mais simples. Mais direcionada também. Achei confuso de entender e, já pensando em aplicar, acho que ler tudo aquilo gasta tempo. Talvez, ter uma lista de ideias (heurísticas) mais direcionadas a tipos de sistema facilitasse o uso da lista.”*

A docente da área de Teste de *Software* também forneceu um depoimento sobre a aplicação do método conduzido em sua turma. Portanto, foi relatado o seguinte: *“É importante observar alguns aspectos no método: (1) Na fase de Execução é deve-se ressaltar que as atividades de execução são feitas em paralelo, e não em separado. As atividades são iterativas. (2) O artefato de planejamento: seria melhor um formulário por funcionalidade. Tive dúvida ao entender seria escopo, no método. Melhor diferenciar bem o que é funcionalidade e escopo, no contexto do método. (3) A checklist de Heurística não foi usada na etapa de execução. (4) O relatório de encerramento está claro. Não houveram dúvidas em relação a este relatório.”*.

Em síntese, após o *feedback* obtido nesta segunda *Validação da Solução* algumas atualizações foram aplicadas ao método *Agile ETeasy*, a citar: (i) as definições dos campos

dos artefatos também foram revisadas e ajustadas para facilitar seu entendimento; (ii) foi incluído no *website* um exemplo real de aplicação do método *Agile ETeasy*, juntamente com o preenchimento dos artefatos; (iii) o vídeo disponibilizado no *website* foi melhorado e reduzido para ficar mais objetivo, em relação à explicação do método; (iv) foi adicionada uma orientação sobre como aplicar a técnica de *Brainwriting*; e, (v) novas orientações sobre conceitos ou definições usados no método, também, foram adicionadas, como a definição de funcionalidade e escopo;

A partir das informações e sugestões fornecidas nesta segunda validação do método *Agile ETeasy*, outras reflexões foram feitas, como: (i) no contexto ágil é comum que a equipe não possua um profissional dedicado exclusivamente aos testes; por este motivo, o método *Agile ETeasy* indica que os testes podem ser executados pelos desenvolvedores ou pelo líder da equipe; (ii) é indicado que o método *Agile ETeasy* seja aplicado sempre ao final do desenvolvimento de uma primeira versão de uma funcionalidade ou de um módulo do sistema implementado no projeto; e, (iii) um trabalho futuro que pode ser desenvolvido é a construção de *Checklists* de Heurísticas direcionadas à tipos de sistemas, como *web*, *Desktop*, *mobile* ou embarcado, para facilitar o uso dessas listas no método.

4.3.4 Implementação e Avaliação da Solução

No intuito de avaliar a implementação do método *Agile ETeasy* do ponto de vista de facilidade de entendimento e utilidade em projetos desenvolvidos no contexto ágil, o Estudo de Caso 2 foi realizado nesta DSR. Assim, após um novo refinamento do método, de acordo com o *feedback* dos avaliadores na etapa de *Validação da Solução* do Ciclo 2, foram planejadas e conduzidas as etapas de *Implementação e Avaliação da Implementação* do método *Agile ETeasy*.

Estas etapas de *Avaliação e Implementação* do método foram realizadas a partir do Estudo de Caso 2 com uma equipe ágil composta por quatro profissionais (desenvolvedores). Assim, o método *Agile ETeasy* foi implementado em um projeto de desenvolvimento de sistemas *web* e ocorreu em dois encontros: um, no dia 23 de junho de 2022; e, o outro, no dia 30 de junho de 2022. A maneira com que os dados foram coletados no Estudo de Caso 2 foi a mesma do Estudo de Caso 1, conforme apresentado no Apêndice G.

Do mesmo modo que no Estudo de Caso 1, foram coletadas algumas informações prin-

cipais referentes à caracterização dos participantes deste estudo. Os dados pessoais dos participantes não foram informados, nesta tese, para garantir o anonimato das informações.

Inicialmente, foram extraídas informações sobre a experiência profissional dos participantes com práticas de Teste de *Software*, no contexto ágil. Foi constatado que todos os participantes possuem o mesmo nível de formação acadêmica, ou seja, todos estão concluindo um curso de graduação em Ciência da Computação, e atuam no mercado de trabalho como desenvolvedores (75% dos participantes) ou *Tech Lead* (25% dos participantes). Em relação ao tempo de experiência profissional, 50% dos participantes possuem 1 ano de experiência com desenvolvimento de *software*, enquanto que 25% deles possuem entre 2 e 3 anos de experiência na mesma área. Outros 25% dos participantes possuem entre 1 e 2 anos de experiência profissional com desenvolvimento de *software* e liderança de equipes ágeis.

Sobre a prática de Teste de *Software*, no contexto ágil, 50% dos participantes informaram ter menos de 1 ano de experiência, enquanto que outros 50% deles informaram ter entre 1 e 2 anos de experiência com testes, em projetos ágeis. Em relação à experiência com testes, os participantes informaram que: “*Possuo experiência em testes manuais em outro projeto*”, “*Já fiz testes básicos para validar requisitos*”, “*Não possuo muita experiência com testes, apenas utilizei testes de unidade e alguns testes exploratórios, de forma empírica*” e “*Realizei testes na disciplina de Engenharia de Software e alguns testes simples*”.

Todos os participantes informaram que possuem o hábito de testar todas as funcionalidades que desenvolvem, com frequência (50% deles) ou esporadicamente (outros 50% dos participantes). No geral, os testes que costumam ser feitos são testes manuais ou testes exploratórios. Em relação à experiência com ET, os participantes informaram que:

- Participante 1: “*Basicamente testei a integração e algumas funcionalidades do nosso sistema submetendo o software a diferentes entradas de acordo com o contexto que usamos.*”
- Participante 2: “*Experiência em outros projetos, neste projeto meu foco maior foi desenvolvimento.*” e “*Tinha o domínio do contexto do projeto e diariamente testava as funcionalidades implementadas. Eu e outra colega eramos direcionadas somente para testes.*”
- Participante 4: “*Já usei ferramentas em disciplinas (da graduação).*” e “*A experiência*

levou a descobrir alguns bugs, que melhoraram a qualidade do produto final.”

Em síntese, percebe-se que o perfil da equipe que participou deste estudo é um perfil de profissionais que tem o hábito de implementar funcionalidades de uma aplicação e testá-las, em seguida. A equipe conhece e já aplicou ET, embora não tenha implementado um método específico. Constatou-se também que a equipe compõe um projeto que integra academia e indústria, e que mesmo ainda não tendo concluído uma graduação, todos os participantes possuem um nível de experiência básico com práticas de teste.

Após a coleta de informações sobre a caracterização dos participantes, do mesmo modo que no Estudo de Caso 1, foi conduzido um mini-curso¹¹ sobre o método *Agile ETeasy*. Este mini-curso foi realizado no primeiro encontro e teve a duração de, aproximadamente, 1 hora e 30 minutos, pois além de apresentar o método, foi feita uma contextualização sobre ET e uma exemplificação de aplicação do método. Em seguida, os participantes foram conduzidos a implementar o método *Agile ETeasy* em um módulo desenvolvido no projeto em que atuam.

Em síntese, a equipe considerou mais adequado planejar os testes em conjunto. Assim, foram definidos oito escopos de teste referentes à uma mesma funcionalidade do projeto. Em seguida, a equipe se dividiu em duplas (como orienta a técnica de *Pair Review*) e iniciaram a etapa de Execução do método *Agile ETeasy*. Uma das duplas executou o teste presencialmente, e a outra dupla, remotamente. Em ambas as duplas, um desenvolvedor executou o teste e narrou a execução, e o outro acompanhou, atentamente, a execução do teste, auxiliando com o registro do tempo de duração do ET, com a verificação e organização dos arquivos usados, com sugestões do que verificar no ET, dentre outras atividades.

Dentre os testes realizados, foram registrados um defeito, duas sugestões de melhorias e quatro anotações de pendências nas funcionalidades. As demais funcionalidades não apresentaram falhas ou inconsistências. No geral, testar essas funcionalidades com o método *Agile ETeasy* durou, aproximadamente, 1 hora e 20 minutos - considerando a implementação das etapas de planejamento, execução e encerramento.

Na discussão feita durante a reunião de alinhamento com toda a equipe - na etapa de Encerramento -, as sugestões de melhorias registradas no relatório foram discutidas e amplificadas em anotações inseridas no campo ‘Notas’ do relatório de encerramento. As discussões

¹¹Acesso ao material adotado no mini-curso: <https://bitly.com/fXJkyknc>.

nesta reunião estimularam reflexões válidas à implementação de melhorias na funcionalidade testada pela equipe.

Em seguida, com o intuito de avaliar a implementação do método *Agile ETeasy* foram aplicados um Diário de Bordo *online* e entrevistas com os participantes do Estudo de Caso 2. A avaliação conduzida nesta etapa de *Avaliação da Implementação* foi importante para mais um refinamento no método e para a percepção sobre a facilidade de entendimento e utilidade do método no contexto ágil.

Após a execução dos testes, cada participante registrou, no seu Diário de Bordo (consultar *template* no Apêndice I), sua experiência com a implementação dos elementos do método *Agile ETeasy*, os pontos positivos, os pontos negativos, os resultados alcançados e/ou as dificuldades encontradas. Ao final do estudo, as informações registradas nos Diários de Bordo foram sintetizadas e analisadas. No geral, não houveram críticas ou sugestões de melhorias a serem implementadas no método *Agile ETeasy*, mas comentários sobre a utilidade de alguns principais elementos. Sobre a atividade “*Definir o escopo de teste*”, da etapa de Planejamento, os participantes informaram que:

- Participante 1: “*Boa estratégia mas dependendo da forma com que você descreve o teste pode ficar ambíguo.*”
- Participante 2: “*Eu particularmente gostei da definição do escopo do teste em pares, pois direciona melhor o que deve de fato ser testado e as prioridades de teste.*”

Ainda, em relação à etapa de Planejamento, os participantes informaram sobre o “artefato de planejamento” que o consideram “*Bem direto e organizado para gerar um caso para ser analisado*”; assim como, a orientação “*Responder a questão: Qual o teste mais importante eu posso fazer agora?*”, em que o participante afirmou “*Achei interessante para definir o que precisa ser testado mais prioritariamente, envolve um entendimento do sistema e um pensamento crítico sobre o mesmo.*”.

Em relação à etapa de Encerramento, os participantes afirmaram que o artefato relatório de encerramento é um “*método bom e o template facilita seu uso*”, “*Cria um foco em apenas um tipo de teste*” e “*A criação de uma documentação para os testes realizados, (de forma padronizada) ajuda muito na hora de criar relatorios de teste*”. Ainda sobre esta etapa, os participantes afirmaram sobre a atividade “*Registrar os testes*”: “*A execução é apenas uma*

formalização e uma documentação para registrar o teste. Importante para discussões junto como time.". Sobre a atividade de "Alinhar as melhorias", os participantes informaram que:

- Participante 1: "Melhor etapa na minha opinião, dá pra esclarecer um conjunto coisas que ficaram 'entre-aberto' durante os testes."
- Participante 2: "Achei de extrema importância, pois alinha o que foi testado e consegue definir os próximos passos."
- Participante 3: "Bastante útil na hora de entender um possível comportamento bastante distoante (em um contexto global)."
- Participante 4: "Achei interessante, pois cada integrante pode ter uma visão diferente para uma melhoria."

Em relação à técnica "Pair Review", um dos participantes informou que "A execução dos testes em pares é um ponto positivo, pois pode ser alinhado diferentes pontos de vistas do que seria o resultado ideal". Sobre a atividade "Observar os resultados", foi informado que "A execução foi tranquila, essa é a parte que envolve mais pensamento crítico juntamente com a etapa de planejamento, pois precisa observar os resultados e comparar com os esperados."

Por fim, um dos participantes registrou no Diário de Bordo sua opinião sobre o método Agile ETeasy: "Como eu já tinha contato com testes exploratórios a execução, seguindo os passos do método foi mais uma formalização. Essa formalização ajuda a manter a organização do que vai ser testado."

Em síntese, foi possível perceber que as informações descritas no Diário de Bordo se referiram a utilidade do método Agile ETeasy e seus pontos fortes na aplicação de ET no contexto dos participantes. Para complementar os resultados desse estudo, foi conduzida uma entrevista individual com os participantes. A síntese das entrevistas pode ser verificada em <https://bitly.com/MfKohuzi>.

Inicialmente, os participantes foram questionados sobre a facilidade de entendimento do método Agile ETeasy, considerando as etapas, atividades e tarefas que o compõem. A seguir, é possível observar a opinião de cada participante do estudo:

- Participante 1: *“Eu achei fácil, razoavelmente fácil. Quando você estava apresentando sobre o método, eu achei que seria mais complexo, mas na prática eu achei mais fácil.”*
- Participante 2: *“Eu, particularmente, gostei do método...E eu achei esse método bem esclarecedor.”*
- Participante 3: *“Eu acho que bem tranquilo, fácil de entender. E aí eu acho que a ideia de entender as etapas é muito bom, até para discutir a formatação da documentação usada. Bem interessante.”*
- Participante 4: *“Com esse método, a gente consegue necessariamente sentar e organizar (o teste), por exemplo... Então, a gente vê quais são as funcionalidades que a gente tem, quais são os problemas que a gente poderia ter e o tipo, o ambiente que a gente tem para poder testar, e acaba auxiliando demais na formalização e até mais no entendimento para posteridade.”*

Em relação às técnicas indicadas no método *Agile ETeasy*, os participantes informaram que já faziam uso das técnicas sem que percebessem que as estavam adotando. Outra ressalva importante, foi sobre a técnica *Pair Review*. Os participantes informaram que *“Eu acho que é importante, sim, em pares, porque teve um determinado teste que o meu entendimento era um, e o entendimento da participante 1, era outro. Então acho que é importante a gente alinhar assim, com pensamentos diferentes, o que seria o ideal.”* (Participante 2) e *“Eu gosto muito dessa parte de ser algo em pares e interessante, tanto programação como reviews e teste em vários. Será muito, muito bom, porque você sempre alguém para discutir alguma coisa relacionada a funcionalidade”* (Participante 3). Um dos participantes ressaltou que a técnica de *Brainwriting* é interessante de ser aplicada, pois se refere à *“criação de uma ideia mais formal, pelo menos mentalmente. E depois é cada vez mais restringindo o escopo para que é possível testar.”* (Participante 4).

Sobre as tarefas indicadas no método *Agile ETeasy*, os participantes relataram que não sentiram dificuldades em compreendê-las. Já em relação aos artefatos de teste gerados nas etapas de Planejamento e Encerramento do método, alguns participantes indicaram a necessidade de incluir uma descrição nos campos, que direcione o profissional a inserir um informação textual mesmo quando não houver informações a serem inseridas, por exemplo

“O que aconteceria se eu não encontrasse um defeito, só colocaria nada ou explicaria o que não foi encontrado no defeito?”, questionou o participante 4.

Em seguida, os participantes foram questionados sobre a utilidade das etapas, atividades e tarefas do método *Agile ETeasy*. A seguir apresentamos as principais observações dos participantes:

- Participante 1: “Eu acho que é útil sim, principalmente para formalizar e registrar (os testes).”
- Participante 2: “Eu acho que essas etapas são importantes, desde o planejamento à execução e, principalmente nessa parte do encerramento, desse feedback que a gente deu em grupo. Eu acho que é muito importante a gente alinhar certos seus pontos de vista também - o que deve ser feito depois da realização do teste. ”
- Participante 3: “Eu acho que é bem útil para para o contexto do que a gente faz agora, principalmente por fazer em conjunto.”
- Participante 4: “São bastante úteis porque como é uma formalização. Então a gente pode mostrar para o nosso gerente, o que a gente fez e o que aconteceu para resolver.”

Sobre a utilidade das técnicas indicadas no método, ambos os participantes informaram que são muito úteis, principalmente a técnica de *Pair Review*, como já afirmado anteriormente. Não houveram sugestões de alteração, adição ou remoção de atividades no método. Entretanto, um dos participantes expôs uma dúvida sobre onde documentar a correção de um *bug* detectado no teste e sugeriu que esta informação fosse inserida nas orientações do método.

Também foi indicada a necessidade de automatização do método por meio de uma ferramenta que facilitasse a geração dos artefatos de testes, principalmente o relatório de encerramento, de acordo com o participante 1: “Automatizar o relatório do encerramento seria interessante... Agilizaria o método. A gente economizaria tempo.”

Dentre as técnicas indicadas para a implementação do método *Agile ETeasy*, os participantes indicaram que só não utilizaram a Pesquisa Documental, pois já tinham conhecimento prévio do produto e da funcionalidade a ser testada. Algumas outras observações adicionais foram fornecidas pelos participantes. Essas informações são apresentadas a seguir:

- Participante 1: *“Eu gostei. Eu tenho uma certa experiência com testes exploratórios em outro contexto, de outro projeto, e eu achei que é muito mais organizado dessa forma, porque geralmente a gente faz os testes sem seguir um padrão. Então, eu acho que essa forma é bem interessante.”*
- Participante 2: *“Tive uma experiência bem positiva em relação a outra experiência que eu tinha tido um teste exploratório. Então eu acho que foi bem positivo em relação às duas que eu tive.”*
- Participante 3: *“O método é intuitivo e bem tranquilo de executar. Não é algo muito complexo... o meu 'maior medo' era isso: tomar muito tempo da gente, porque a gente está com um tempo apertado. Mas, basicamente, é bem tranquilo e bem mais efetivo. À medida que a gente vai testando, a gente vai fazendo de forma mais rápida... Ultimamente eu estou tentando implementar essa melhora na minha equipe, principalmente essa formalização de todas as coisas.”*

Diante dessas informações coletadas e do que foi gerado como oportunidades de melhorias, é possível afirmar que o método *Agile ETeasy* conseguiu auxiliar os profissionais participantes deste Estudo de Caso 2, na execução de Testes Exploratórios, demonstrando ser fácil de entender e útil ao contexto das equipes.

Do mesmo modo que nas etapas empíricas anteriores, o Estudo de Caso 2 forneceu algumas sugestões de melhorias que serviram para o refinamento do método e foram aplicadas da seguinte forma: (i) foi adicionada uma orientação de preenchimento dos artefatos para quando não houver uma informação a ser inserida nos campos; e, (ii) foi adicionada uma orientação sobre a realização de um novo teste, sempre que um *bug* detectado na funcionalidade for corrigido, ou seja, não é indicado que o *bug* corrigido seja testado novamente e registrado no mesmo relatório de encerramento, pois se trata de um novo teste.

Por fim, também foi sugerida a implementação de uma ferramenta para automatização dos artefatos de teste. Essa sugestão é considerada como um trabalho futuro a ser desenvolvido a partir desta tese.

4.4 Considerações Finais do Capítulo

Este capítulo explanou detalhadamente os procedimentos adotados no Ciclos da *Design Science Research* implementada nesta pesquisa. Com base na proposta de Wieringa [78] foram implementados dois ciclos de DSR, organizados de acordo com as seguintes etapas: *Investigação do Problema, Design da Solução, Validação da Solução, Implementação da Solução e Avaliação da Implementação*. Para facilitar a compreensão desta metodologia e melhor organizar este capítulo, cada Ciclo da DSR foi descrito em uma seção.

A Seção 4.1 descreveu os procedimentos metodológicos adotados para auxiliar na compreensão do problema investigado, na construção do artefato e no refinamento do artefato proposto. Assim, foi apresentado o planejamento para cada etapa executada no Ciclo 1 e no Ciclo 2 desta DSR, de modo a destacar os estudos desenvolvidos em cada ciclo.

A Seção 4.2 apresentou os resultados das etapas de *Investigação do Problema, Design da Solução, Validação da Solução e Implementação da Solução*.

A Seção 4.3 discutiu os resultados da etapa de *Avaliação da Implementação, Design da Solução, Validação da Solução e Implementação da Solução*.

Por fim, o próximo capítulo discutirá alguns trabalhos relacionados à implementação de Testes Exploratórios no contexto de desenvolvimento de *software* em ambientes ágeis.

Capítulo 5

Trabalhos Relacionados

Para apoiar os pesquisadores e os profissionais ágeis na realização de atividades de Testes Exploratórios, a pesquisa desenvolvida nesta tese foi embasada, também, na investigação de estratégias que apoiam a implementação e o uso de ET no cotidiano de trabalho de equipes ágeis. Assim, este Capítulo apresenta uma relação entre as contribuições desta tese e o estado da arte publicado na literatura sobre a implementação de Testes Exploratórios por equipes ágeis, que atuam na indústria de desenvolvimento de *software*. Os trabalhos aqui discutidos são resultantes de uma RSL conduzida na etapa de *Investigação do Problema* do Ciclo 1 da DSR (conforme apresentado na Seção 4.2.1). Portanto, discutimos os resultados dos principais trabalhos relacionados a esta temática, na Seção 5.1. Em seguida, com base em uma análise comparativa dos trabalhos relacionados, apresentamos algumas conclusões na Seção 5.2, que nos permitem ratificar a relevância desta pesquisa. E, por fim, expomos as conclusões deste Capítulo, na Seção 5.3.

5.1 Testes Exploratórios no contexto ágil

Alguns estudos têm se dedicado à investigação de ET no contexto de indústria com o intuito de perceber o impacto ou a eficácia dessa prática de teste em projetos reais que adotam método ágil [2; 35; 48; 55; 62; 70].

Geibzli and Sozer [35] avaliaram o impacto da educação e o nível de experiência dos testadores sobre a eficácia do ET. Para isso foi realizado um estudo de caso com 19 profissionais da indústria, com diferentes formações educacionais e níveis de experiência. Um

sistema de TV Digital foi testado, e as falhas detectadas foram categorizadas de acordo com a sua gravidade. Assim, a eficácia do ET foi avaliada sobre dois aspectos: criticidade das falhas detectadas e eficiência no número de falhas detectadas por unidade de tempo. Os resultados mostram que a eficiência do ET é significativamente afetada pela formação e pela experiência educacional.

Martensson et al. [55] realizou um estudo baseado em entrevistas para compreender os fatores de sucesso na aplicação de ET em projetos da indústria. Para isso, foram realizadas entrevistas com 20 profissionais. Por fim, foi apresentada uma lista de fatores-chave que permitem a eficiência e eficácia de ET em sistemas de *software* em larga escala. Os nove fatores identificados são agrupados em quatro temas: (i) O conhecimento, a experiência e a personalidade dos testadores; (ii) Objetivo e escopo; (iii) Formas de trabalho; e, (iv) Registro e relatório.

Pfahl et al. [62] investigou como os engenheiros de *software* entendem e aplicam o princípio de ET, bem como as vantagens e dificuldades que vivenciam. Para isso, foi realizada uma pesquisa *online* entre desenvolvedores de *software* estonianos e finlandeses, e testadores. Os principais resultados indicam que a maioria dos testadores, desenvolvedores e gerentes de teste que usam ET, (1) aplicam ET a *softwares* críticos para usabilidade, desempenho, e segurança em alto grau; (2) utilizam o ET com muita flexibilidade em todos os tipos de níveis, atividades e fases; (3) percebem o ET como uma abordagem que apoia a criatividade durante o teste, e é eficaz e eficiente; e, (4) consideram que o ET não é fácil de usar e que existem poucas ferramentas de apoio. Além disso, foi constatada a necessidade de mais apoio aos usuários de ET, como orientações e ferramentas.

Afzal et al. [2] buscou quantificar a eficácia e eficiência do ET vs. Testes com Casos de Teste documentados (TCT). Para isso, foram realizados quatro experimentos controlados, com um total de 24 profissionais e 46 alunos. Para isso, foram executados testes funcionais manuais, usando ET e TCT. Foram medidos o número de defeitos identificados nas sessões de teste de 90 minutos, a dificuldade de detecção, gravidade e tipos dos defeitos detectados e o número de relatórios de defeitos falsos. Os resultados mostram que o ET encontrou um número significativamente maior de defeitos. No entanto, as duas abordagens de teste não diferiram expressivamente, em termos do número de relatórios de defeitos falsos.

Souza et al. [70] investigou o uso de ET em aplicativos móveis. Para isso, foram

realizados dois estudos. O primeiro estudo objetivou aplicar ET a aplicativos com contextos diversos e disponíveis no *Google Play*, a fim de analisar se os testadores realmente explorariam todos os cenários possíveis que os aplicativos podem exibir. O segundo estudo, também aplicou o ET, porém em dois aplicativos que foram desenvolvidos por uma empresa de desenvolvimento de *software*, com o objetivo de identificar *bugs* de diferentes níveis, que muitas vezes não podem ser percebidos por outras técnicas de teste. Os resultados indicam, para o primeiro estudo, que existem vários cenários de teste que não são explorados pelos testadores, mas os 40 participantes revelaram em média 5 *bugs* em 1,5h de sessões de teste; já, para o segundo estudo, foram percebidos 64 *bugs* e 21 problemas nos dois aplicativos. Em síntese, o ET tem se mostrado uma técnica promissora para descobrir *bugs*, embora os profissionais de teste possam ser melhor orientados para explorar os aplicativos de dispositivos móveis.

Kedziora et al. [48] examinou a adequação de ET no processo de Teste de Software. Para isso, foi conduzido um experimento, em duas partes. Primeiramente, foi realizado um teste e, na segunda parte, foi realizado um levantamento, que permitiu a comparação de ET e baseados em testes. Os resultados indicam uma eficácia um pouco menor da abordagem exploratória, o que pode ter sido causado por condições do experimento, como escolha do *software* testado, curta duração das sessões de teste, falta de conhecimento dos participantes sobre o *software* investigado e experiência na realização de ET. Entretanto os ET mostraram-se úteis, principalmente na detecção de erros distintos, não encontrados durante testes baseados em casos de teste. No experimento, 90% dos entrevistados confirmaram o uso da abordagem de teste formalizada, baseada em casos de teste, enquanto pouco mais da metade (57%) indicou ter experiência na realização de ET.

5.1.1 Suporte à execução de ET na indústria ágil

Embora a literatura evidencie algumas abordagens ou propostas de *frameworks* para a aplicação de Teste Exploratório no contexto de desenvolvimento de *software* ágil, ainda são poucos os estudos que apresentam resultados consolidados ou se referem a um processo completo e bem definido, aplicável ao contexto real de equipes ágeis [9; 36; 81; 53; 65].

Jiu Jiu et al. [81] analisa as limitações dos testes ágeis atuais, por meio de um novo

modelo de Teste Exploratório de *software* ágil, combinando ET e testes automatizados. A ideia de *design* deste modelo é baseada no *framework Agile Scrum*. Inicialmente, cada teste é realizado em cada fase do ciclo de iteração e adaptado aos recursos de desenvolvimento ágil. Em seguida, além de completar os testes de funcionalidade e desempenho, é feita a verificação do processo de desenvolvimento de *software*. Nesse modelo, não apenas os testadores devem participar de todo o processo do *Scrum*, mas também de todo o processo de teste: na preparação do teste, no pré-teste, no teste de funcionalidade e no teste de regressão, que devem ser executados em um ciclo iterativo. O resultado do experimento mostra que, comparando os testes ágeis com os tradicionais, mais *bugs* na eficácia e usabilidade do sistema podem ser detectados nos mesmos ciclos de teste e, conseqüentemente, melhores resultados de teste são obtidos. Embora Jiu Jiu et. al. [81] tenha afirmado que o uso *framework Agile Scrum* tenha sido eficaz na detecção de *bugs*, este modelo não contempla especificamente o processo de aplicação do ET e não explica como ele ocorre para o ET.

Basri et al. [9] descreve uma proposta de *framework* para Testes Exploratórios, a ser adotado no contexto de Pequenas e Médias Empresas de *software* (PMEs). Inicialmente, a proposta consiste em analisar e discutir, por meio de uma Revisão de Literatura, as lacunas de pesquisa existentes em estudos sobre Testes Exploratórios. Deste modo, vários processos de teste e outros fatores associados à qualidade de *software* são analisados. Em seguida, o *framework* é proposto, com o objetivo de melhorar a qualidade do *software* desenvolvido por profissionais de PMEs. Embora o *framework* aponte alguns benefícios interessantes ao contexto de PMEs, ainda não foi aplicado um estudo experimental para validar características como facilidade de entendimento, facilidade de uso, eficácia ou eficiência, dentre outros aspectos, com profissionais de PMEs.

Ghazi et. al. [36] apresenta uma *Checklist* para apoiar a descrição das Cartas de Teste em ET, com o objetivo de apoiar os profissionais que adotam esta estratégia para a execução de ET. Deste modo, buscou-se identificar fatores que permitissem aos profissionais refletir criticamente sobre seus projetos e conteúdos de Cartas de Teste, com o intuito de apoiar estes profissionais na tomada de decisões sobre o que incluir nas Cartas de Teste. Os fatores e conteúdos foram levantados por meio de nove entrevistas. As entrevistas foram utilizadas para reunir uma lista de verificação para os fatores que influenciam o *design* da Cartas de Teste e outra para descrever o possível conteúdo destas cartas. No geral, 30 fatores e 35

elementos de conteúdo foram identificados e categorizados. Dentre esses fatores podemos destacar: requisitos do cliente, estratégia de teste, conhecimento de *bugs* anteriores, áreas de risco, dentre outros. Dentre os elementos, podemos citar: configuração do teste, foco do teste, nível do teste, técnicas de teste, riscos, *bugs* encontrados, dentre outros. Mesmo com uma lista detalhada de fatores e elementos que contribuem para a elaboração de Cartas de Teste, percebemos que este tipo de artefato não é completamente aplicável ao contexto de desenvolvimento ágil, devido principalmente ao nível de detalhes e precedentes que o testador deve conhecer para definir uma Carta de Teste.

Rapaana et al. [65] relata como o uso de uma abordagem de Teste Exploratório em Equipe (TET) afeta os resultados nos testes de *software*. O estudo foi realizado na *FSecure Corporation*, onde dois projetos foram investigados. Os resultados mostram que as sessões de TET apresentam boa eficácia e eficiência superior a outros métodos de teste na empresa medidos em número de defeitos detectados. A comparação do tipo de defeito detectado revelou que as sessões TET são mais propensas a detectar defeitos relacionados à usabilidade e à interface do usuário, enquanto outros meios de teste detectam mais funcionalidades relacionadas. Os participantes da sessão viram benefícios principalmente na discussão conjunta e no aprendizado da aplicação, e destacam a necessidade de mais pesquisas para estudar os efeitos do Teste Exploratório em Equipe (TET). Em síntese, esse estudo enfatiza uma abordagem baseada na integração contínua da equipe para executar o ET. Entretanto, com a rigidez dos papéis definidos para uma sessão de TET, existe o risco de algum membro da equipe não possuir a habilidade ou experiência indicada para o teste, ou ainda não possuir o conhecimento sobre o domínio do sistema a ser testado. Esses fatores são cruciais para a eficácia do TET.

Leveau et al. [53] apresenta uma nova abordagem para ajudar os testadores a explorar amplamente qualquer aplicação *web*, por meio dos ET. Em particular, esta abordagem monitora as interações *online* realizadas pelos testadores para sugerir em tempo real as probabilidades de realizar as próximas interações. Esta abordagem define um modelo de previsão, baseado em *n*-gramas, que codifica o histórico de interações passadas e que suporta a estimativa das probabilidades. Integrado em uma extensão de um navegador *Web*, ele gera um *feedback* de forma automática e transparente no próprio aplicativo. Para avaliar esta abordagem foi realizado um experimento controlado e um estudo qualitativo. Os resultados

mostram que esta abordagem evita que os testadores fiquem presos em *loops* já testados e também consegue auxiliá-los na realização de explorações mais profundas do *System Under Test* (SUT). Entretanto, esta abordagem é direcionada apenas à aplicações *web* - aspecto que dificulta seu uso mais abrangente, para outros tipos de sistemas e em outros contextos de projetos ágeis.

5.2 Posicionamento desta pesquisa em relação aos trabalhos relacionados

Percebemos que dentre os trabalhos encontrados na literatura de *Agile ST*, poucos estudos têm investigado a implementação de ET em projetos ágeis da indústria de desenvolvimento de *software*. Bem como, a inserção de práticas já consolidadas ou a criação de novas abordagens que orientam a execução dos ET neste contexto. A Tabela 5.1 apresenta uma síntese dos onze trabalhos discutidos neste Capítulo. As colunas consideradas para a construção da tabela são as seguintes:

- **Trabalho:** Autores e referência bibliográfica do trabalho relacionado à esta pesquisa.
- **Tipo da pesquisa:** Tipo de pesquisa aplicada no trabalho.
- **Abordagem de ET:** Definição da abordagem de Teste Exploratório usada no trabalho.
- **Projetos reais:** Tipo de sistema ou de projeto de *software* real utilizado na pesquisa.
- **Profissionais ágeis:** Aplicação da pesquisa com profissionais que atuam com métodos ágeis.

Em seguida, na Tabela 5.2 é exposta uma análise comparativa desta pesquisa com os trabalhos que apresentam abordagens para ET, discutidos na Seção 5.1.1 deste Capítulo. Para isso, alguns desafios comuns aos ET foram levantados [33; 71; 77], a fim de perceber quais destes desafios foram solucionados ou mitigados com a implementação da abordagem proposta nos referidos trabalhos. Esses desafios são explicados a seguir e incluídos como colunas da referida Tabela. Estão destacados com “X” os desafios que foram solucionados ou mitigados pelo método apresentado nesta tese. A caracterização desta pesquisa, em relação

às demais, é apresentada na última linha. Assim, as colunas consideradas para a construção da tabela são as seguintes:

- **Pouca Experiência do Testador:** A abordagem buscou diminuir a dependência da experiência e habilidades do testador;
- **Auto-gerência e Gerência da equipe:** A abordagem apoiou a integração entre os membros da equipe durante o teste.
- **Falta de Foco da equipe:** A abordagem adotou alguma estratégia para que a equipe ou o testador mantenham o foco na atividade de teste, enquanto a realiza.
- **Desconhecimento do Domínio da Aplicação:** A abordagem indicou alguma orientação ou meio para que o testador tenha uma ideia aproximada do que testar no domínio da aplicação.
- **Documentação dos Testes:** A abordagem propiciou alguma forma de registro ou preparação dos testes.
- **Aplicação de Testes Exploratórios a Sistemas de Tempo Real:** A abordagem indicou alguma estratégia para a realização do ET.
- **Reprodutibilidade dos Testes:** A abordagem indicou ou disponibilizou alguma estratégia para preparação do ET, a fim de garantir a consistência dos resultados, caso o teste seja realizado novamente.

Nesse sentido, ainda não foi encontrada uma abordagem validada e aceita pela indústria de *software*, compatível com as reais necessidades das equipes ágeis. As abordagens existentes, como já mencionado, foram desenvolvidas e utilizadas em estudos específicos e contextos isolados, considerando: o envolvimento de alguns profissionais, nas entrevistas; dados de projetos reais de desenvolvimento de *software*, em experimentos controlados; e, em alguns poucos casos, experiências com equipes ágeis.

Portanto, esta pesquisa atende 6 dos 7 desafios expostos na Tabela 5.2. Este trabalho se diferencia dos demais por: (i) apresentar uma proposta de um método mais completo, do que apenas orientações para definição de cartas de teste. Foi percebido, neste estudo, que

Cartas de Teste convencionais, ou com muitos detalhes, não facilitam a execução do ET; *(ii)* apresentar um método baseado em etapas, tarefas, técnicas e artefatos mais aplicáveis ao contexto de equipes que atuam com desenvolvimento ágil, construído com base em validações iterativas e contínuas com profissionais ágeis; *(iii)* o método proposto pode ser aplicado em qualquer tipo de sistema, seja ele *web*, *desktop*, *mobile* ou embarcado. Esse aspecto permite que o método seja usado para testar diferentes contextos de projetos ágeis; *(iv)* se basear em um conjunto de atividades que orientam o planejamento e a execução do ET, por meio do uso da técnica *Pair Review*; dentre outros aspectos discutidos ao longo dos próximos capítulos.

Tabela 5.1: Síntese dos trabalhos relacionados à esta pesquisa.

| Trabalho | Tipo da pesquisa | Abordagem de ET | Projetos reais | Profissionais ágeis |
|------------------------|--------------------------------|--|--|---------------------|
| Geibzli and Sozer [35] | Estudo de caso | Não aplicada | Sistema de TV Digital | Sim |
| Martensson et al. [55] | Entrevistas | Não aplicada | Sistemas de <i>software</i> em grande escala | Sim |
| Pfahl et al. [62] | Survey | Não aplicada | Não identificado | Sim |
| Afzal et al. [2] | Experimento lado | SBTM | Não identificado | Sim |
| Souza et al. [70] | Estudo de caso | Não aplicada | Aplicativos Móveis | Sim |
| Kedziora et al. [48] | Experimento e Entre- vistas | Não aplicado | Não identificado | Sim |
| Jiu Jiu et al. [81] | Experimento | Análise comparativa de técnicas | Não aplicado | Não |
| Basri et al. [9] | Estudo descritivo | <i>Framework</i> para ET | Não aplicado | Não |
| Ghazi et. al. [36] | Entrevistas | <i>Checklist</i> para Car- tas de Teste | Não aplicado | Sim |
| Rapaana et al. [65] | Estudo de caso | TET | <i>FSecure Corporation</i> | Sim |
| Leveau et al. [53] | Experimento e Qualitativo | Monitoramento de interações online | Aplicações <i>web</i> | Sim |

Tabela 5.2: Análise comparativa das abordagens destacadas nos trabalhos relacionados.

| | Jiu Jiu et al. [81] | Basri et al. [9] | Ghazi et. al. [36] | Rapaana et al. [65] | Leveau et al. [53] | Esta pesquisa |
|--|---------------------|------------------|--------------------|---------------------|--------------------|---------------|
| Pouca Experiência do Testador | | X | | X | | X |
| Auto-gerência e Gerência da equipe | X | | | | | X |
| Falta de Foco da equipe | | | | X | X | X |
| Desconhecimento do Domínio da Aplicação | | | | | X | |
| Documentação dos Testes | | | | | | X |
| Aplicação de ET a Sistemas de Tempo Real | | | | | | X |
| Reprodutibilidade dos Testes | | | | | | X |

5.3 Considerações finais do Capítulo

Neste Capítulo foram apresentados alguns dos principais trabalhos relacionados à área de pesquisa de *Agile ST*. O propósito desta Seção foi expor os trabalhos encontrados em um levantamento bibliográfico, com objetivo de discutir pesquisas relevantes sobre ET no contexto ágil da indústria de *software*.

A Seção 5.1 apresentou uma síntese dos principais trabalhos que investigam ET em projetos de desenvolvimento de *software* ágil. Em seguida, também foram discutidos trabalhos que propõem e analisam abordagens que visam apoiar a execução dos ET.

A Seção 5.2 traz um sumário dos trabalhos relacionados ao tema de pesquisa desta tese, e uma análise comparativa entre os trabalhos que apresentam abordagens que apoiam a aplicação de ET em projetos desenvolvidos por equipes ágeis. Em seguida, foi discutido, brevemente, um posicionamento desta pesquisa em relação aos trabalhos listados.

Por fim, o próximo capítulo discutirá as considerações finais desta tese, ressaltando as conclusões obtidas ao final da pesquisa, as principais contribuições, as ameaças a validade, principais limitações e algumas propostas de trabalhos futuros.

Capítulo 6

Considerações Finais

Neste capítulo são apresentadas as conclusões finais desta tese, de acordo com os resultados obtidos com a implementação da DSR. As conclusões da pesquisa são discutidas na Seção 6.1. Em seguida, algumas principais contribuições da pesquisa são apresentadas na Seção 6.2. Depois disso, as ameaças a validade e as principais limitações do estudo são destacadas na Seção 6.3 e na Seção 6.4. E, por fim, são apresentadas possibilidades de trabalhos futuros a serem desenvolvidos a partir desta tese, na Seção 6.5.

6.1 Conclusões

Esta pesquisa de doutorado identificou duas lacunas na área de *Agile ST* referentes à aplicação de Testes Exploratórios no contexto de equipes ágeis: (i) a ausência de um suporte mais específico para o planejamento e execução do ET, considerando o perfil multifuncional das equipes ágeis; e, (ii) a dificuldade em registrar os ET, devido à ausência de artefatos robustos. Assim, a principal Questão de Pesquisa (QP) investigada nesta tese foi: “*O que fazer para conseguir integrar Testes Exploratórios, como prática de teste, no cotidiano de equipes ágeis, considerando um cenário de requisitos que não são formalmente documentados e que mudam constantemente?*”.

Nesse sentido, o objetivo dessa pesquisa de doutorado foi apoiar a implementação de Testes Exploratórios em ambientes de desenvolvimento de *software* que adotam práticas de métodos ágeis, que não possuem requisitos documentados formalmente e a mudança nos requisitos é frequente, por meio da implementação de um método que considera as principais

boas práticas de ET. Como diferencial, esse método adaptou as fases de planejamento e execução comumente adotadas por outras abordagens de ET (como SBTM e *Charters de Teste* [6; 36]) para se adequarem ao perfil multifuncional das equipes e construiu artefatos de testes que contribuem com o registro dos ET aplicados.

Para responder a QP principal desta pesquisa, foram definidas QPs específicas (conforme descrito na Seção 1.4). A seguir são apresentados os principais resultados gerados para cada questão de pesquisa.

QP1. Quais são as principais abordagens, práticas, ferramentas e artefatos existentes que podem auxiliar os ET aplicados no contexto ágil?

QP2. Quais são as principais limitações de aplicar ET no ambiente ágil?

As equipes de desenvolvimento de *software* têm se concentrado no desenvolvimento de produtos que visam atender as necessidades dos clientes, em um cenário de constantes mudanças nos requisitos, de pouca ou nenhuma documentação e de clientes ativos no projeto.

Buscando responder a QP1 e a QP2, foram realizadas na etapa de *Investigação do Problema* do Ciclo 1 alguns estudos para identificar as principais abordagens, práticas, ferramentas e artefatos existentes que apoiam a prática de ET, no contexto ágil. De modo complementar, também foram investigadas as principais limitações na implementação de ET, por equipes ágeis.

De acordo com a RSL conduzida nesta DSR foi constatado que poucas abordagens e práticas de ET foram usadas no contexto prático com equipes ágeis [9; 36; 53; 65; 81]. A Seção 4.2.1 discute as principais abordagens e práticas encontradas. De acordo com a RSL realizada foi constatado que (i) ET é uma área de estudo ainda pouco investigada, principalmente quando se refere ao ambiente ágil; (ii) a literatura recente indica poucas abordagens de ET usadas na prática; e, (iii) estudos empíricos com equipes ágeis são necessários para validar a aplicação prática de alguns abordagens existentes.

Da mesma forma foram investigadas as principais ferramentas existentes que apoiam a execução de ET. O apêndice B apresenta a lista das dezoito (18) ferramentas identificadas neste estudo. Foi percebido que a ferramenta que mais se aproxima às necessidades de gerenciamento de ET é a *XRay Exploratory App*. As demais caracterizam-se mais por fun-

cionalidades de gravação e de *report* de dados. Tais aspectos tornam essas ferramentas úteis a um contexto geral de aplicação do ET, pois ao analisar as especificidades que compõem o contexto de gerenciamento de ET (consultar 4.2) foi constatado que boa parte das ferramentas não são eficazes. Outro aspecto percebido é ausência de uma funcionalidade que aponte o requisito testado.

Com a realização das entrevistas com os líderes de equipes que atuam com desenvolvimento de *software* ágil, foi constatado que há pouca demanda e pouco tempo dedicado aos testes, em virtude do prazo atribuído às *sprints* dos projetos. Embora testes manuais demandem um tempo considerável para sua realização e seja uma tarefa repetitiva, ainda são considerados muito úteis nos projetos. Outro aspecto verificado é que os testes geralmente não são planejados porque não há tempo hábil para esta tarefa. Em um equipe ágil, também não há um profissional com *expertise* em ST, principalmente, em ET, e que se dedique exclusivamente aos testes. É importante destacar que durante as entrevistas foi levantada a necessidade de uma abordagem simples de aplicar; que não seja burocrática, em termos de documentação; e, que seja robusta, para testar as aplicações desenvolvidas nos projetos. Mais detalhes podem ser consultados na Seção 4.2.1.

Desse modo, diante das informações descobertas nas entrevistas e considerando os benefícios do ET no desenvolvimento ágil (constatado na RSL), foi percebida a necessidade de integrar a teoria e a prática em ET para um melhor entendimento dos efeitos desse tipo de teste no ambiente ágil. Os resultados indicam que o curso sobre ET facilitou a compreensão dos conceitos e das práticas de ET pelos participantes. A existência de uma ferramenta de apoio para ET também foi fundamental para otimizar o aprendizado, no contexto remoto. Em síntese, tanto as práticas como os artefatos foram bem aproveitados na execução do teste. Entretanto, mesmo com as orientações fornecidas no curso, algumas limitações foram percebidas como a ausência de um apoio mais específico para o planejamento e execução do ET, como orientações e ferramentas. Por isso, para confirmar a existência dessas limitações em um contexto prático e real de desenvolvimento de *software*, os participantes foram motivados a aplicar sessões de ET no seu contexto de trabalho (mais detalhes são fornecidos na Seção 4.2.2).

QP3. Como organizar a maneira de executar ET de modo a otimizar sua prática no cenário ágil?

A resposta para essa questão de pesquisa envolve a construção do método *Agile ETeasy* a partir dos resultados obtidos na etapa de *Investigação do Problema* do Ciclo 1 e dos refinamentos realizados nas etapas de *Validação* e *Avaliação da Implementação* do Ciclo 1 e 2 da DSR, conforme descritos no Capítulo 4.

A utilização da DSR como procedimento metodológico contribuiu significativamente para o *design* e a implementação do método *Agile ETeasy*, fornecendo orientações base para a identificação do problema, o *design* da solução e as sucessivas validações e avaliações implementadas, o que possibilitou o refinamento do método sempre que uma nova validação ou avaliação foi conduzida.

Como pode ser visto na Figura 4.3 o método *Agile ETeasy* é composto por etapas, atividades, tarefas, técnicas e elementos de saída que conduzem a aplicação de Testes Exploratórios por profissionais de equipes ágeis, que atuam em um contexto onde os requisitos não são formalmente documentados e mudam constantemente. As Seções 3.1, 3.2 e 3.3 descrevem detalhadamente os elementos que compõem o método *Agile ETeasy*, e fornecem orientações gerais para sua implementação.

Em síntese, o método *Agile ETeasy* visa cobrir algumas limitações do ET no contexto ágil, como a pouca experiência da equipe com práticas de teste, a falta de foco da equipe, a documentação e a reprodutibilidade dos testes [33], por meio da indicação do uso de uma *checklist* de heurísticas, de testes feitos em par, de artefatos de teste baseados em uma planilha de execução e a indicação de leitura de documentos de referência, caso necessário para a aplicação do teste [33].

Os resultados da implementação do método *Agile ETeasy* obtidos por meio dos Estudos de Caso 1 e 2 [80], foram descritos na Seção 4.3.1 e na Seção 4.3.4. No geral, esses resultados indicam que o método *Agile ETeasy*, após os sucessivos refinamentos aplicados, demonstrou ser organizado, intuitivo, fácil de entender e útil ao contexto ágil dos profissionais que participaram dos estudos.

De modo complementar, alguns benefícios iniciais da implementação do método *Agile ETeasy* puderam ser observados ao longo dos Estudos de Caso aplicados, como economia de tempo no planejamento de *scripts* de teste, detecção rápida de falhas no sistema, testes

registrados com mais produtividade, diminuição de artefatos de teste e melhor compreensão de funcionamento do produto pelo testador.

6.2 Contribuições

Abordagens para apoiar o gerenciamento e a medição dos ET são indicadas e discutidas na literatura de *Agile ST*. Entretanto, embora tais abordagens tenham se mostrado eficientes na aplicação do ET, algumas limitações são percebidas quando aplicadas na prática por equipes ágeis. Associado a essa problemática, ainda são poucos os estudos que investigam a adoção de ET ou discutem formas de tornar mais prática a implementação de ET em projetos ágeis.

Dessa forma, como principal contribuição desta tese está o *design* e a implementação de um método que apoia a execução de Testes Exploratórios em cenários de desenvolvimento de *software* que adotam métodos ágeis. Como mencionado no Capítulo 3, o método é denominado *Agile ETeasy* e é composto por três etapas: a etapa de Planejamento, que consiste na definição do escopo do teste a ser executado; a etapa de Execução, que se refere à exploração do escopo de teste; e, a etapa de Encerramento, que consiste no registro e discussão do comportamento da funcionalidade verificada no teste. Espera-se que, com esse método, equipes ágeis sejam capazes de:

- Implementar Testes Exploratórios em seu cotidiano de trabalho, de modo simples e objetivo, seguindo um conjunto de etapas que formaliza a utilização do ET sem burocracias adicionais.
- Gerar artefatos de Testes Exploratórios, simples e robustos, que garantam o fácil registro e reprodutibilidade dos testes, posteriormente.

Outros benefícios gerais da adoção do método *Agile ETeasy* também são considerados, como: a mitigação das principais limitações da aplicação de ET em projetos que adotam métodos ágeis; e, a formalização prática de ET, que consideram o contexto multifuncional das equipes ágeis, e que fornece resultados rápidos, com pouca quantidade de recursos.

Desse modo, as contribuições produzidas nesta tese procuram preencher a lacuna gerada pela ausência de um apoio mais específico para o planejamento e execução do ET, considerando o perfil multifuncional de uma equipe ágil; por artefatos de ET obsoletos para o

contexto atual das equipes ágeis; pela dificuldade em reproduzir os ET, devido à ausência de artefatos robustos; e, pela falta de evidências empíricas atuais sobre a aplicação de abordagens de ET no contexto prático de equipes ágeis.

Nesse sentido, o Capítulo 3 apresentou a construção do método *Agile ETeasy* (como contribuição científica) contendo as etapas, atividades, tarefas, técnicas e artefatos de teste. Enquanto que, a Seção 4.2.4 e a Seção 4.3.1 apresentou os resultados da implementação do método (como contribuição prática) no contexto de equipes ágeis.

Como contribuições adicionais, o desenvolvimento desta pesquisa contribuiu com as seguintes publicações científicas:

- 2019 - **COUTINHO, J.C.S.**; ANDRADE, W. L.; MACHADO, P. D. L. Requirements Engineering and Software Testing in Agile Methodologies: a Systematic Mapping. In: 33rd Brazilian Symposium on Software Engineering (SBES 2019), 2019, Salvador - BA. SBES 2019, 2019 (Artigo completo).
- 2021 - **COUTINHO, J.C.S.**; ANDRADE, W. L.; MACHADO, P. D. L. Teaching Exploratory Tests through PBL and JiTT: an experience report in a context of distributed teams. In: 35th Brazilian Symposium on Software Engineering (SBES 2021), 2021, Joinville - Brazil. SBES 2021, 2021 (Artigo completo).
- 2022 - **COUTINHO, J.C.S.**; ANDRADE, W. L.; MACHADO, P. D. L. A Survey of Requirements Engineering and Software Testing Practices in Agile Teams. In: 7th Brazilian Symposium on Systematic and Automated Software Testing (SAST 2022), 2022, Uberlandia - Brazil. SAST 2022, 2022 (Artigo completo).

É importante destacar que, outros trabalhos derivados desta pesquisa foram submetidos para publicação ou estão sendo refinados para submissão em periódicos.

6.3 Ameaças a validade

No decorrer do *design* e da execução desta DSR, notou-se que alguns aspectos caracterizaram-se como ameaças à validade desta pesquisa, conforme a classificação apresentada por Wohlin et al. [79]: ameaças a validade interna, externa, de *constructo* e de conclusão. Por esse motivo, algumas iniciativas foram adotadas para mitigá-las.

A **validade de constructo** se refere à fatores que podem surgir em decorrência de um *design* impróprio do instrumento de pesquisa. Para mitigar a *validade de constructo*, todas as atividades desenvolvidas nas etapas da DSR, que culminaram no *design* e construção do método *Agile ETeasy*, foram planejadas, atualizadas e validadas iterativamente pela autora deste tese e por mais dois profissionais experientes na área de Teste de *Software*. Bem como, foram embasados em trabalhos relacionados à área de ET [78; 6; 71; 77; 26; 33].

A **validade interna** está relacionada a fatores que definem a forma como os resultados observados são equivalentes a realidade, e portanto, não se podem introduzir um erro sistemático ou viés nos resultados. Na pesquisa qualitativa a realidade é construída e interpretada. Desse modo, para garantir a mitigação da validade interna nesta pesquisa qualitativa foram adotados diferentes instrumentos de pesquisa para coleta de dados, como, questionários semi-estruturados, diários de bordo, entrevistas e brainstorming. Para mitigar a ameaça a validade interna, além de se aplicar diferentes instrumentos de pesquisa, também foi considerada a participação de profissionais com diferentes perfis. Adicionalmente, para evitar o viés nos resultados e garantir o anonimato das respostas fornecidas, a identificação dos participantes do estudo foi opcional - por meio do nome e endereço de *email*.

A **validade externa** está relacionada ao quanto os resultados da pesquisa são extensivos a outras situações ou contextos. Desse modo, já existe uma perda de validade externa ao selecionar os participantes, desta pesquisa, por conveniência, tanto na etapa de *Validação da Solução* como na etapa de *Implementação da Solução* do Ciclo 1. Apesar de haver uma maior participação de profissionais na etapa de *Validação da Solução* e de *Implementação da Solução* do Ciclo 2, os resultados obtidos se referem apenas a experiência daqueles profissionais e ao contexto daquelas equipes. No entanto, para mitigar esta ameaça a validade da pesquisa foram apresentadas as particularidades dos contextos em que o método *Agile ETeasy* foi validado e implementado, de modo a fornecer detalhes sobre o refinamento do método, que podem guiar o leitor a aproximar ou verificar se os resultados da pesquisa são aplicáveis ao seu contexto.

A **validade de conclusão** é percebida quando medidas estatísticas impróprias ou erros em testes conduzem a conclusões incorretas. Considerando a pesquisa qualitativa conduzida nesta tese, para mitigar a validade de conclusão, foi adotada a *Grounded Theory* que contribuiu tanto na análise, como na representação textual dos dados. Em relação aos dados

quantitativos coletados nos questionários semi-estruturados, foram adotadas, apenas porcentagens para identificar padrões comuns. Adicionalmente, também foram descartadas as respostas da validação do questionários elaborados, quanto a possíveis erros, como: expressões textuais, formato das respostas, questões redundantes, dentre outros.

6.4 Limitações

Apesar do rigor metodológico planejado e conduzido, de acordo com a DSR aplicada nesta tese, a execução e os resultados desta pesquisa apresentam algumas limitações. Assim, as limitações percebidas são descritas a seguir, de acordo com a implementação do Ciclo 1 e do Ciclo 2 desta DSR - conforme discutido, respectivamente, na Seção 6.4.1 e na Seção 6.4.2.

É importante ressaltar que por se caracterizar como uma pesquisa qualitativa, não foi objetivo desta tese obter resultados generalizáveis. Por este motivo, mesmo com a implementação de dois ciclos da DSR, contendo a contribuição de 17 profissionais especialistas nas etapas de *Validação da Solução* e com a implementação do artefato no contexto de trabalho de duas equipes ágeis, considera-se que o método *Agile ETeasy* ainda precisa ser aplicado em contexto diferentes, por meio de novos estudos experimentais com profissionais ágeis.

6.4.1 Limitações da implementação do Ciclo 1 da DSR

Algumas limitações foram percebidas na implementação do Ciclo 1 da DSR. Portanto, destacamos: (i) na etapa de *Investigação da Solução*, a limitação do levantamento de práticas de teste realizadas nos projetos desenvolvidos por equipes ágeis, consistiu na quantidade de participantes entrevistados. Considerando, que estes participantes não costumavam aplicar ET nos projetos, não foi possível obter um *insight* inicial sobre as desvantagens ou dificuldades desta prática de teste. Deste modo, foi necessário ampliar a busca de informações e aplicar um curso de ET para que tais informações pudessem ser levantadas. Nesta mesma etapa, foi constatada outra limitação referente (ii) à baixa quantidade de trabalhos empíricos atuais sobre a implementação de ET no contexto de equipes ágeis - o que também limitou a observação de resultados que validam a utilização de outras abordagens de ET. Por fim, outra limitação foi observada na (iii) na etapa de *Implementação da Solução*, pois esta etapa envolveu apenas a contribuição de uma equipe ágil composta por 3 profissionais especia-

listas, o que faz com que o grau de generalização dos resultados seja baixo. Outro fator a ser considerado foi a ausência de mais funcionalidades, no projeto, a serem testadas com o método *Agile ETeasy*. Entretanto, por esses motivos, um segundo ciclo da DSR foi aplicado.

6.4.2 Limitações da implementação do Ciclo 2 da DSR

No Ciclo 2, também foram percebidas algumas limitações na realização das etapas, como: (i) na etapa de *Validação da Solução*, pois esta etapa envolveu apenas a contribuição de 12 profissionais especialistas e de um profissional acadêmico, o que faz com que o grau de generalização dos resultados seja baixo. Ressalta-se que foram enviados 51 *emails* para profissionais próximos e, em seguida, ampliou-se o envio para profissionais indicados por terceiros, por meio do método de amostragem não probabilística *snowball sampling*. Outra limitação percebida foi na etapa de *Implementação e Avaliação da Solução*, pois o Estudo de Caso 2 foi aplicado apenas com (ii) uma equipe ágil composta por 4 profissionais, o que faz, mais uma vez, com que o grau de generalização dos resultados seja baixo. Também não houveram outras funcionalidades a serem testadas no projeto, por meio do método *Agile ETeasy*. Por esses motivos, se faz necessário a realização de outros estudos para ampliar e generalizar os resultados.

6.5 Trabalhos Futuros

Com a finalização desta tese e como continuidade desta pesquisa, vislumbram-se as seguintes propostas de trabalhos futuros:

1. O desenvolvimento de uma ferramenta que automatize a implementação do método *Agile ETeasy*, por meio dos artefatos de teste gerados - o artefato de planejamento e o relatório de encerramento -, de modo a facilitar a sua utilização em projetos desenvolvidos por equipes ágeis.
2. Realizar novos estudos experimentais para avaliação da implementação do método *Agile ETeasy* com uma amostra representativa de equipes ágeis a fim de aumentar o poder de generalização e compreender a eficácia do método para a execução de ET no contexto ágil.

3. Avaliar a qualidade dos artefatos de teste gerados após a implementação do método *Agile ETeasy*.
4. Investigar estratégias que considerem o gerenciamento de requisitos e o seu impacto na execução do método *Agile ETeasy*, de modo a fornecer *insights* de melhorias ao método.
5. Validar a implementação de testes automatizados E2E (*End-to-End*) a partir dos artefatos gerados no método *Agile ETeasy*.
6. Uma proposta de *checklist* de heurísticas para ET aplicáveis à tipos de sistemas *software* específicos, como sistemas *web*, *mobile*, *desktop* e embarcados.
7. Desenvolver variações do método *Agile ETeasy* com a inclusão ou remoção de etapas, atividades, tarefas, técnicas ou artefatos de teste, de modo que se adeque a um contexto específico estudado.
8. Validar as demais abordagens de ET, discutidas na Seção 2.4 , em um contexto ágil a fim de perceber suas contribuições e limitações para o planejamento e a execução de ET.
9. Realizar um estudo comparativo entre as abordagens de ET, citadas na Seção 2.4, e o método *Agile ETeasy*, em um ontexto prático.

Bibliografia

- [1] Manifesto for agile development, 2001.
- [2] Wasi Afzal, Ahmad Nauman Ghazi, Juha Itkonen, Richard Torkar, Anneliese Andrews, and Khurram Bhatti. An experiment on the effectiveness and efficiency of exploratory testing. In *Empirical Softw. Engg.*, volume 20, page 844–878, 2015.
- [3] Wasim Alsaqaf, Maya Daneva, and Roel Wieringa. Quality requirements challenges in the context of large-scale distributed agile: An empirical study. *Information and software technology*, 110:39–55, 2019.
- [4] J Angara, S Prasad, and G Sridevi. The factors driving testing in devops setting-a systematic literature survey. *Indian Journal of Science and Technology*, 9(48):1–8, 2017.
- [5] Fredrik Asplund. Exploratory testing: Do contextual factors influence software fault identification? *Information and Software Technology*, 107:101–111, 2019.
- [6] James Bach. Exploratory testing explained, 2003.
- [7] Sergio Luis Barbieri. *Teste de software na indústria: um estudo qualitativo*. PhD thesis, Universidade de São Paulo.
- [8] Samera Obaid Barraood, Haslina Mohd, and Fauziah Baharom. A comparison study of software testing activities in agile methods. In *Knowledge Management International Conference (KMICe) 2021*, 2021.
- [9] Shuib Basri, Dhanapal Durai Dominic, Thangiah Murugan, and Malek Ahmad Almo-mani. A proposed framework using exploratory testing to improve software quality in

- sme's. In Faisal Saeed, Nadhmi Gazem, Fathey Mohammed, and Abdelsalam Busalim, editors, *Recent Trends in Data Science and Soft Computing*, pages 1113–1122, Cham, 2019. Springer International Publishing.
- [10] Shuib Basri, Thangiah Murugan, and Dhanapal Durai Dominic. Validating the conceptual framework with exploratory testing. In Radek Silhavy, editor, *Software Engineering Methods in Intelligent Algorithms*, pages 305–317, Cham, 2019. Springer International Publishing.
- [11] Kent Beck. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.
- [12] Antonia Bertolino. Software testing research: Achievements, challenges, dreams. In *2007 Future of Software Engineering*, pages 85–103. IEEE Computer Society, 2007.
- [13] Elizabeth Bjarnason and Helen Sharp. The role of distances in requirements communication: a case study. *Requirements Engineering*, 22(1):1–26, 2017.
- [14] Elizabeth Bjarnason, Michael Unterkalmsteiner, Markus Borg, and Emelie Engström. A multi-case study of agile requirements engineering and the use of test cases as requirements. *Information and Software Technology*, 77:61–79, 2016.
- [15] Nathalie Bonnardel and John Didier. Brainstorming variants to favor creative design. *Applied ergonomics*, 83:102987, 2020.
- [16] R. Bortoluci and M. Duduchi. Indicadores para teste de software em desenvolvimento ágil. In *X Workshop de Pós-Graduação e Pesquisa do Centro Paula Souza*, pages 764–773, São Paulo, Brazil, 2015.
- [17] Adnan Causevic, Daniel Sundmark, and Sasikumar Punnekkat. Factors limiting industrial adoption of test driven development: A systematic review. In *Fourth IEEE International Conference on Software Testing, Verification and Validation*, pages 337–346, Berlin, Germany, 2011. IEEE.
- [18] Alistair Cockburn. *Crystal Clear a Human-Powered Methodology for Small Teams*. Addison-Wesley Professional, first edition, 2004.

- [19] Eliane Figueiredo Collins and Vicente Ferreira de Lucena. Software test automation practices in agile development environment: An industry experience report. In *Automation of Software Test (AST), 2012 7th International Workshop on*, pages 57–63, Zurich, Switzerland, 2012. IEEE.
- [20] Rubens Copche, Mariana Souza, Isabel Karina Villanes, Vinicius Durelli, Marcelo Eler, Arilo Claudio Dias-Neto, and Andre Takeshi Endo. *Exploratory Testing of Apps with Opportunity Maps*. Association for Computing Machinery, New York, NY, USA, 2021.
- [21] Juliet Corbin and Anselm Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.
- [22] Igor Ernesto Ferreira Costa and Sandro Ronaldo Bezerra Oliveira. A study on assets applied in exploratory test design and execution: an interview application. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, 2021.
- [23] Jarbele Coutinho, Wilkerson Andrade, and Patricia Machado. *Teaching Exploratory Tests through PBL and JiTT: An Experience Report in a Context of Distributed Teams*, page 205–214. Association for Computing Machinery, New York, NY, USA, 2021.
- [24] Jarbele C. S. Coutinho, Wilkerson L Andrade, and Patrícia D. L. Machado. Requirements engineering and software testing in agile methodologies: a systematic mapping. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pages 322–331, Salvador, BA, Brazil, 2019. ACM.
- [25] Adalberto Nobiato Crespo, Odair J Silva, Carlos Alberto Borges, Clênio Figueiredo Salviano, M Argollo, and Mario Jino. Uma metodologia para teste de software no contexto da melhoria de processo. *Simpósio Brasileiro de Qualidade de Software*, pages 271–285, 2004.
- [26] Lisa Crispin and Janet Gregory. *Agile testing: A practical guide for testers and agile teams*. Pearson Education, 2009.
- [27] Nigel Cross. Design/science/research: developing a discipline. In *Proceedings of the Korea Society of Design Studies Conference*, pages 16–24. Korea Society of Design Science, 2001.

- [28] Daniela Soares Cruzes, Michael Felderer, Tosin Daniel Oyetoyan, Matthias Gander, and Irdin Pekaric. How is security testing done in agile teams? a cross-case analysis of four software teams. In *International Conference on Agile Software Development*, pages 201–216, Cologne, Deutschland, 2017. Springer.
- [29] Karina Curcio, Tiago Navarro, Andreia Malucelli, and Sheila Reinehr. Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, 139:32–50, 2018.
- [30] Ernani César dos Santos and Patrícia Vilain. Automated acceptance tests as software requirements: An experiment to compare the applicability of fit tables and gherkin language. In *International Conference on Agile Software Development*, pages 104–119, Porto, Portugal, 2018. Springer.
- [31] Aline Dresch, Daniel P. Lacerda, and José A. V. A. Jr. *Design Science Research - A Method for Science and Technology Advancement*. Springer, 2015.
- [32] Scott Duncan. Dsdm, business focused development. *Software Quality Professional*, 6(3):41, 2004.
- [33] Igor Ernesto Ferreira Costa and Sandro Ronaldo Bezerra Oliveira. An evidence-based study on the efficiency and efficacy of exploratory testing. In *International Conference on Information Systems Technology Management (CONTECSI)*, 2019.
- [34] Caliane de O Figuerêdo, Simone C dos Santos, PH Borba, and GH Alexandre. Using pbl to develop software test engineers. In *International Conference on Computers and Advanced Technology in Education*, pages 305–322. sn, 2011.
- [35] Ceren Şahin Gebizli and Hasan Sözer. Impact of education and experience level on the effectiveness of exploratory testing: An industrial case study. In *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 23–28, 2017.
- [36] Ahmad Nauman Ghazi. *Structuring Exploratory Testing Through Test Charter Design and Decision Support*. PhD thesis, Blekinge Tekniska Högskola, 2017.

- [37] Jo Erskine Hannay, Karsten Brathen, and Ole Martin Mevassvik. Agile requirements handling in a service-oriented taxonomy of capabilities. *Requirements Engineering*, 22(2):289–314, 2017.
- [38] V. T. Heikkilä, D. Damian, C. Lassenius, and M. Paasivaara. A mapping study on requirements engineering in agile software development. In *41st Euromicro Conference on Software Engineering and Advanced Applications*, pages 199–207, Madeira, Portugal, Aug 2015.
- [39] Theodore D. Hellmann and Frank Maurer. Rule-based exploratory testing of graphical user interfaces. In *2011 Agile Conference*, pages 107–116, 2011.
- [40] Elisabeth Hendrickson. *Explore it!: reduce risk and increase confidence with exploratory testing*. Pragmatic Bookshelf, 2013.
- [41] Alan R Hevner. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4, 2007.
- [42] Rashina Hoda, Norsaremah Salleh, John Grundy, and Hui Mien Tee. Systematic literature reviews in agile software development: A tertiary study. *Information and Software Technology*, 85:60 – 70, 2017.
- [43] Irum Inayat, Sabrina Marczak, Siti Salwah Salim, and Daniela Damian. Patterns of collaboration driven by requirements in agile software development teams. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 131–147. Springer, 2017.
- [44] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, and Shahaboddin Shamshirband. A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, 51:915–929, 2015.
- [45] Jingdong Jia, Xiaoying Yang, Rong Zhang, and Xi Liu. Understanding software developers’ cognition in agile requirements engineering. *Science of Computer Programming*, 178:1–19, 2019.
- [46] Cem Kaner and James Bach. Exploratory testing in pairs. In *Software Testing Analysis & Review Conference (STAR) West*, 2001.

- [47] Cem Kaner, Jack Falk, and Hung Q Nguyen. *Testing computer software*. John Wiley & Sons, 1999.
- [48] M. Kedziora, K. KalWak, I. Jozwiak, and M. Szczepanik. *Analysis of exploration testing approach and conclusions from implementation in software development*. Scientific papers of Silensian University of Technology, 2020.
- [49] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1):7–15, 2009.
- [50] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. 2007.
- [51] Gaurav Kumar and Pradeep Kumar Bhatia. Comparative analysis of software engineering models from traditional to modern methodologies. In *2014 Fourth International Conference on Advanced Computing Communication Technologies*, pages 189–196, 2014.
- [52] M. Käpyaho and M. Kauppinen. Agile requirements engineering with prototyping: A case study. In *IEEE 23rd International Requirements Engineering Conference (RE)*, pages 334–343, Ottawa, Canada, Aug 2015.
- [53] Julien Leveau, Xavier Blanc, Laurent Réveillère, Jean-Rémy Falleri, and Romain Rouvoy. Fostering the diversity of exploratory testing in web applications. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, pages 164–174, 2020.
- [54] N. Maia, G. Macedo, E. Collins, and A. D. Neto. Aplicando testes ágeis com equipes distribuídas: Um relato de experiência. In *Simpósio Brasileiro de Qualidade de Software*, pages 365–372, Fortaleza, CE, Brazil, 2012.
- [55] Torvald Mårtensson, Antonio Martini, Daniel Ståhl, and Jan Bosch. Excellence in exploratory testing: Success factors in large-scale industry projects. In *Product-Focused Software Process Improvement*, pages 299–314. Springer International Publishing, 2019.

- [56] Juliana Medeiros, Daniela C. P. Alves, Alexandre Vasconcelos, Carla Silva, and Eduardo Wanderley. Requirements engineering in agile projects: A systematic mapping based in evidences of industry. In *Proceedings of the XVIII IberoAmerican Conference on Software Engineering, CIbSE 2015*, page 460, 2015.
- [57] Juliana Medeiros, Alexandre Vasconcelos, Miguel Goulão, Carla Silva, and João Araújo. An approach based on design practices to specify requirements in agile projects. In *Proceedings of the Symposium on Applied Computing*, pages 1114–1121, Marrakech, Morocco, 2017. ACM.
- [58] Juliana Medeiros, Alexandre Vasconcelos, Carla Silva, and Miguel Goulão. Quality of software requirements specification in agile projects: A cross-case analysis of six companies. *Journal of Systems and Software*, 142:171 – 194, 2018.
- [59] Gregor M Novak. Just-in-time teaching. *New directions for teaching and learning*, 2011(128):63–73, 2011.
- [60] Mirosław Ochodek and Sylwia Kopczyńska. Perceived importance of agile requirements engineering practices—a survey. *Journal of Systems and Software*, 143:29–43, 2018.
- [61] Steve R. Palmer and Mac Felsing. *A Practical Guide to Feature-Driven Development*. Pearson Education, 1st edition, 2001.
- [62] Dietmar Pfahl, Huishi Yin, Mika V. Mäntylä, and Jürgen Münch. How is exploratory testing used? a state-of-the-practice survey. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '14*, New York, NY, USA, 2014. Association for Computing Machinery.
- [63] Andrew Pham and Phuong-Van Pham. *SCRUM em ação*. Novatec Editora, 2011.
- [64] Roger S Pressman. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.
- [65] Paula Raappana, Soili Saukkoriipi, Ilkka Tervonen, and Mika V Mäntylä. The effect of team exploratory testing—experience report from f-secure. In *2016 IEEE Ninth*

- International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 295–304. IEEE, 2016.
- [66] Fabio Gomes Rocha, Rogério Patrício Chagas do Nascimento, and Methanias Colaço Rodrigues Junior. A survey on agile methodologies and its relationship with the requirement-based techniques. 2017.
- [67] Soili Saukkoriipi and Ilkka Tervonen. Team exploratory testing sessions. *International Scholarly Research Notices*, 2012:20, 2012.
- [68] Eva-Maria Schön, Jörg Thomaschewski, and María José Escalona. Agile requirements engineering: A systematic literature review. *Computer Standards Interfaces*, 49:79–91, 2017.
- [69] Ian Sommerville. *Software Engineering*. Addison-Wesley, 2011.
- [70] Mariana Souza, Isabel K. Villanes, Arilo Claudio Dias-Neto, and Andre T. Endo. On the exploratory testing of mobile apps. In *Proceedings of the IV Brazilian Symposium on Systematic and Automated Software Testing, SAST 2019*, page 42–51, New York, NY, USA, 2019. Association for Computing Machinery.
- [71] Beni Suranto. Exploratory software testing in agile project. In *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, pages 280–283. IEEE, 2015.
- [72] Sulabh Tyagi, Ritu Sibal, and Bharti Suri. Adopting test automation on agile development projects: A grounded theory study of indian software organizations. In *International Conference on Agile Software Development*, pages 184–198, Cologne, Deutschland, 2017. Springer.
- [73] Eric Verzuh, American Psychological Association, et al. A guide to the project management body of knowledge: Pmbok guide. 2021.
- [74] A. Vicente, M. E. Delamaro, and J. C. Maldonado. Uma revisão sistemática sobre a atividade de teste de software em métodos Ágeis. In *Proceedings of the XXXV Conferencia Latinoamericana de Informática*, Pelotas, RS, Brazil, 2009.

-
- [75] Stefan Wagner, Daniel Méndez Fernández, Michael Felderer, Antonio Vetrò, Marcos Kalinowski, Roel Wieringa, Dietmar Pfahl, Tayana Conte, Marie-Therese Christiansson, Desmond Greer, et al. Status quo in requirements engineering: A theory and a global family of surveys. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 28(2):1–48, 2019.
- [76] Stefan Wagner, Daniel Méndez Fernández, Michael Felderer, and Marcos Kalinowski. Requirements engineering practice and problems in agile projects: results from an international survey. *XX Ibero-American Conference on Software Engineering (CIbSE)*, *arXiv:1703.08360*, 2017.
- [77] James A Whittaker. *Exploratory software testing: tips, tricks, tours, and techniques to guide test design*. Pearson Education, 2009.
- [78] Roel J Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.
- [79] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [80] Robert K Yin. *Estudo de Caso-: Planejamento e métodos*. Bookman editora, 2015.
- [81] Jiujiu Yu. Design and application on agile software exploratory testing model. In *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pages 2082–2088, 2018.

Apêndice A

Apêndice A - Protocolo da Revisão Sistemática da Literatura (RSL)

De acordo com Kitchenham e Charters [49], o desenvolvimento de uma Revisão Sistemática da Literatura (RSL) deve seguir passos metodologicamente bem definidos, englobando três fases: o planejamento, a condução e a publicação de pesquisa. Assim, esta RSL tem como intuito realizar uma busca por estudos primários sobre Testes Exploratórios (ET) desenvolvidos no ambiente ágil. Para isso, é proposta a análise dos estudos primários a partir de critérios de exclusão e inclusão, critérios de qualidade, e, por fim, respostas à questões de pesquisa.

A.1 Questões de Pesquisa investigadas

Buscamos investigar a seguinte questão de pesquisa: “*Como ET tem sido conduzido, recentemente, no contexto ágil?*”. Deste modo, para auxiliar na resposta desta questão de pesquisa, foram definidas algumas Questões Específicas (QE) de pesquisa:

QE_1 : Quais estratégias de gerenciamento de ET têm sido usadas para apoiar a execução de ET no ambiente ágil?

QE_2 : Quais os benefícios de aplicar ET em projetos de desenvolvimento de *software* no contexto ágil?

QE_3 : Quais as limitações de aplicar ET em projetos de desenvolvimento de *software* no contexto ágil?

A.2 Busca por trabalhos primários

Para a realização da busca por trabalhos primários foram selecionadas cinco bases de dados: *ACM Digital Library*, *Google Scholar*, *IEEE Explore*, *Science Direct* e *Springer Link*. De modo adicional, os anais de algumas conferências também foram analisados, como: *Brazilian Symposium on Systematic and Automated Software Testing (SAST)*, *Brazilian Symposium on Software Engineering (SBES)*, *Brazilian Symposium on Software Quality (SBQS)*, *International Conference on Software Testing, Verification and Validation (ICST)*, *International Conference on Information Systems Engineering (ICISE)*, *Symposium on Applied Computing (SAC)* e *International Conference on Software Engineering (ICSE)*.

- *ACM Digital Library*: <https://dl.acm.org/>
- *IEEE Xplore*: <https://ieeexplore.ieee.org/Xplore/home.jsp>
- *Google Scholar*: <https://scholar.google.com.br>
- *Scienc Direct*: <https://www.sciencedirect.com/>
- *Springer Link*: <https://link.springer.com/>

A busca por trabalhos primários ocorreu em dois momentos distintos. A primeira pesquisa por trabalhos primários foi aplicada, em sua primeira vez, no período de setembro à dezembro de 2020, em busca de trabalhos publicados no período de 2016 à 2020. E, mais tarde, atualizada no período de janeiro à abril de 2022, em busca de trabalhos publicados nos anos de 2021 e meados de 2022.

Para realizar a busca foi definida a seguinte *String* de busca: “*exploratory testing*” and “*exploratory testing tools*” and “*tools for exploratory testing*” and “*software and project*”. Esta *string* foi aplicada na 1ª busca por trabalhos em todas as bases de dados supracitadas. E, também na 2ª busca na base de dados da ACM. Entretanto, a *string* precisou ser modificada para a 2ª busca nas demais bases de dados, em virtude da quantidade de trabalhos primários retornados ao aplicá-la inicialmente. Portanto, na 2ª busca, nas demais bases de dados foi aplicada a *string*: (“*exploratory testing*” and “*software and project*”).

A.3 Critérios de Inclusão e Exclusão

Para a seleção dos trabalhos primários foram definidos os seguintes Critérios de Inclusão (CI):

- **CI1:** Os trabalhos devem estar disponíveis para *download*.
- **CI2:** Trabalhos que descrevem pesquisas de caráter experimental ou teórico.
- **CI3:** Trabalhos que se referem ao contexto da pesquisa.
- **CI4:** Trabalhos publicados entre 2016 até 2020, e mais tarde, 2021 até meados de 2022.
- **CI5:** Trabalhos desenvolvidos no contexto ágil.

Também foram definidos os seguintes Critérios de Exclusão (CE):

- **CE1:** Trabalhos que não estejam disponíveis para consulta ou download (em versão completa).
- **CE2:** Trabalhos que não contém as Strings de Busca da pesquisa em seu título, resumo ou nas palavras-chave.
- **CE3:** Trabalhos repetidos encontrados em mais de uma Base de Dados. Neste caso, considerar apenas o de um das bases.
- **CE4:** Trabalhos do tipo: resumos, *keynote*, palestras, cursos, tutoriais, *workshops* e afins.
- **CE5:** Trabalhos incompletos (menos de 6 páginas).
- **CE6:** Trabalhos que sejam escritos em outro idioma, exceto o inglês.

A.4 Critérios de Qualidade

Após a aplicação dos critérios de inclusão e exclusão, foi avaliado a qualidade dos estudos primários resultantes da fase anterior. A avaliação da qualidade foi realizada com os estudos

primários incluídos nesta fase. Assim, a qualidade dessa RSL foi avaliada de acordo com os critérios de qualidade [49]. Assim, foram utilizados quatro Critérios de Qualidade (CQ) em forma de perguntas, a citar:

CQ₁: O estudo primário descreve a metodologia de pesquisa empregada?

CQ₂: O estudo destaca pelo menos uma prática, estratégia, ferramenta ou técnica adotada em Testes Exploratórios?

CQ₃: Os resultados apresentados referem-se a estudos realizados em cenários reais de equipes ágeis?

CQ₄: Os dados/resultados foram claramente descritos?

Todos os estudos primários foram classificados com base em quão bem eles atenderam aos critérios de qualidade. O seguinte sistema de pontos foi utilizado para determinar os escores individuais: Sim (S) = 1 ponto, Parcial (P) = 0.5 pontos, Não (N) = 0 ponto. A pontuação geral de qualidade foi calculada pela soma da pontuação relatada nos quatro critérios individuais. Assim, o escore total de qualidade de cada estudo variou de 0 (muito ruim) a 4 (muito bom).

A.5 Seleção dos Trabalhos

Para o processo de filtragem dos trabalhos retornados na busca realizada nas bases de dados, foram definidas algumas etapas a serem seguidas (consultar Figura A.1). A primeira etapa é caracterizada pela pré-seleção, que consiste em realizar a busca por trabalhos nas bases de dados por meio da *string* de busca definida. Em seguida, foi aplicada a primeira filtragem dos trabalhos, por meio da análise e aplicação dos CE1, CE2 e CE6 no título, palavras-chave e o resumo de cada trabalho. Depois disso, foi aplicado a segunda filtragem, por meio dos CE3, CE4 e CE5 na Introdução e Conclusão dos trabalhos. Por fim, foi feita a leitura completa dos trabalhos e aplicados os critérios de inclusão, CI1, CI2, CI3 e CI4. Os artigos pré-selecionados foram analisados de acordo com os critérios de qualidade estabelecidos neste protocolo. E, finalmente, os trabalhos primários relacionados ao contexto desta pesquisa foram extraídos.



Figura A.1: Etapas seguidas nesta RSL.

Apêndice B

Apêndice B - Ferramentas para Teste

Exploratório

Neste Apêndice são apresentadas as dezoito (18) ferramentas identificadas por meio de uma revisão bibliográfica realizada na etapa de *Investigação do Problema* do Ciclo 1 desta DSR, e complementada por meio de uma pesquisa manual realizada na *web*, usando palavras-chave como “ferramenta” e “testes exploratórios”. A Tabela B.1 apresenta o nome da ferramenta, o endereço *web* de acesso e a referência à um endereço *web* que disponibiliza um tutorial de uso da ferramenta.

Tabela B.1: Endereços *web* de acesso às ferramentas selecionadas.

| Ferramenta | Acesso (<i>download</i>) | Tutorial de Uso |
|-----------------------------------|---|---|
| <i>Azure Test Plans</i> | https://azure.microsoft.com/en-us/services/devops/test-plans/ | https://docs.microsoft.com/en-us/azure/devops/test/?view=azure-devops |
| <i>BB Assistant</i> | https://www.flashbackrecorder.com/testassistant/ | https://www.flashbackrecorder.com/files/Flashback_5_User_Guide.pdf |
| <i>Bug Magnet</i> | https://bugmagnet.org/ | https://bugmagnet.org/using.html |
| <i>Exploratory Testing</i> | https://github.com/morvader/ExploratoryTestingChromeExtension | https://chrome.google.com/webstore/detail/exploratory-testing-chrom/khigmghadjljgjpamimgjjmpmlbgmekj |
| <i>Jira + Jira Capture</i> | https://www.atlassian.com/br/software/jira/capture | https://cutt.ly/IfIr5Aa |
| <i>Practi Test</i> | https://www.practitest.com/ | https://marketplace.atlassian.com/apps/1211223/practitest-test-management-for-jira?hosting=cloud&tab=installation |
| <i>QMetry Exploratory Testing</i> | https://chrome.google.com/webstore/detail/qmetry-for-jira-qtm4j-exp/caailpilkjenpnbidcbbhmpgobcgfo | https://www.qmetry.com/resources/videos/ |

Continua na próxima página

Tabela B.1 – Continua na próxima página

| Ferramenta | Acesso (<i>download</i>) | Tutorial de Uso |
|-----------------------------|--|--|
| <i>Qtest Explorer</i> | https://www.tricentis.com/products/agile-dev-testing-qtest/exploratory-testing-qtest-explorer/ http://testing.gershon.info/reporter/ | https://www.tricentis.com/academy/training/?tri-ondemand-type=58#archive-results http://testing.gershon.info/wp-content/uploads/Rapid-Reporter-Readme.pdf |
| <i>Rapid Reporter</i> | https://www.inflectra.com/Rapise/ | https://www.inflectra.com/Rapise/Documentation.aspx |
| <i>SmartBear Zephyr</i> | https://www.getzephyr.com/ | https://zephyrdocs.atlassian.net/wiki/spaces/ZFJ0500/overview |
| <i>Spira Capture</i> | https://www.inflectra.com/SpiraCapture/ | https://spiradoc.inflectra.com/SpiraCapture/User-Guide/ |
| <i>Testpad</i> | https://ontestpad.com/ | https://ontestpad.com/examples |
| <i>Testrail</i> | https://www.gurock.com/testrail/ | https://www.gurock.com/testrail/docs/user-guide |
| <i>Test Studio Explorer</i> | https://docs.telerik.com/teststudio/features/testing-types/exploratory-testing/launch-tool | https://www.telerik.com/videos/teststudio |

Continua na próxima página

Tabela B.1 – Continua na próxima página

| Ferramenta | Acesso (<i>download</i>) | Tutorial de Uso |
|----------------------|---|---|
| <i>Teststuff</i> | https://www.teststuff.com/ | https://www.teststuff.com/support/video-tutorials/ |
| <i>Wink</i> | https://www.debugmode.com/wink/ | https://cutt.ly/xfU846A |
| <i>XRay Explorer</i> | https://www.getxray.app/ | https://docs.getxray.app/display/XEA/ |
| <i>App</i> | exploratory-testing-app/ | Getting+Started |

Apêndice C

Apêndice C - Entrevista realizada na *Investigação do Problema do Ciclo 1*

I - Apresentação do entrevistado:

1. Você pode falar um pouco sobre sua formação acadêmica? (Graduação ou pós, Universidade)
2. Você pode falar um pouco sobre sua experiência de mercado? (área de atuação, tempo, tipos de projetos)
3. Você conhece métodos ágeis? Utiliza algum específico com sua equipe? E como acontece no trabalho remoto?

II - Sobre a equipe:

4. Sobre o projeto que está trabalhando atualmente: Qual sua função na equipe? O que você faz (suas responsabilidades)? Você pode descrever sua equipe? (quantos membros e quais as responsabilidades de cada um) Como as reuniões acontecem? Usam alguma estratégia de métodos ágeis? Qual?

III - Sobre o sistema:

5. Qual o tipo de aplicação desenvolvida por sua equipe? (*Web, desktop, mobile*)
6. Como extraem as informações do que deve ser desenvolvido? (ferramenta, pessoa, frequência)
7. Vocês registram essas informações? Como é feito esse registro? Quem é o responsável?
8. Essas informações (extraídas e registradas) são suficientes?

9. Como essas informações são controladas, quando ocorre alguma mudança?

IV - Sobre a atividade de teste:

10. Existe alguém responsável pelos testes, em sua equipe? Quem é a pessoa?

11. Em que momento os testes são feitos?

12. Qual tipo de teste de *software* é executado?

13. Como esses testes são executados? Como fazem, atualmente? (Você pode detalhar essa atividade? - ferramenta, pessoa, momento, etc.)

14. Essa estratégia de teste tem funcionado bem? Porque você a considera adequada?

15. Você sente alguma dificuldade ao realizar os testes (ausência de: uma formação complementar, um modelo/método que auxilie na execução, uma ferramenta, outro)?

16. Você pode detalhar um pouco mais essas dificuldades?

17. Alguma estratégia ágil é usada para auxiliar nos testes?

V - Encerramento da Entrevista:

18. Você deseja nos dar mais alguma informação que considere importante, e que não contemplamos nesta entrevista?

19. Ou, deseja dizer suas impressões sobre a entrevista.

Apêndice D

Apêndice D - Protocolo do Experimento sobre SBTM

Neste Apêndice é apresentado o protocolo adotado na realização do Experimento sobre o uso da abordagem SBTM por profissionais que integram equipes ágeis de desenvolvimento de *software*. Uma contextualização é apresentada na Seção D.1. Em seguida, é exposto o planejamento do experimento na Seção D.2. Depois disso, é descrita a execução do experimento na Seção D.3. E, por fim, são discutidos os procedimentos de análise na Seção D.4.

D.1 Visão Geral

Dentre as práticas de ST existentes, temos os Testes Exploratórios (*Exploratory Testing - ET*) que enfatizam a responsabilidade e a liberdade do testador para explorar o sistema, permitindo que o testador adquira conhecimento do programa em paralelo a execução dos testes [6; 33; 40; 77], pois não há o planejamento de *scripts* ou a definição de casos de testes, definidos em planos de testes [40]. Para Bach [6], o ET é aprendizagem, projeto e execução de testes realizados de maneira simultânea.

Como forma de suprir a necessidade de gerenciamento e medição do teste exploratório, Bach [6] propôs (1) dividir as atividades de testes em sessões, que seriam a unidade básica de trabalho, (2) estipular uma missão para cada sessão e (3) adotar métricas do tempo relacionado às atividades de teste [6], originando assim a abordagem *Session-Based Test Management* (SBTM).

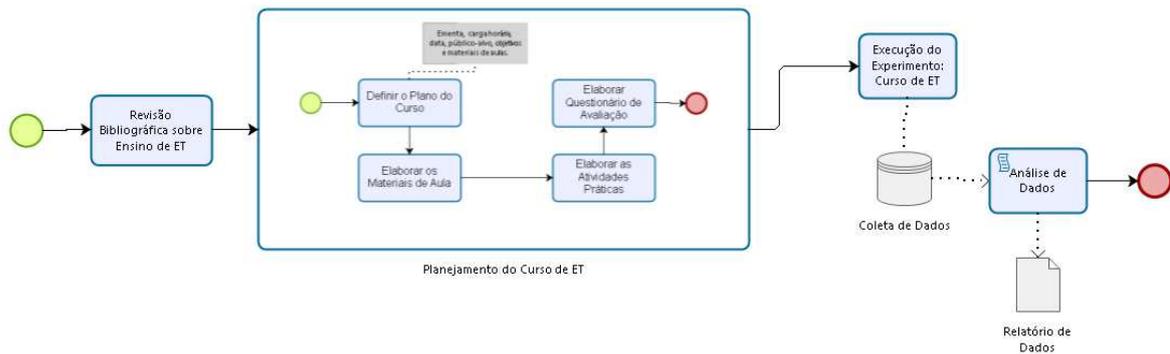


Figura D.1: Fases de desenvolvimento deste experimento.

Assim, o objetivo desse experimento é investigar como o SBTM pode contribuir para a incorporação de Testes Exploratórios em projetos de desenvolvimento de *software* no contexto ágil.

Em síntese, esta pesquisa é classificada como um relato de experiência [79], pois descreve precisamente: o planejamento, a execução e os procedimentos de análise como forma de contribuir com considerações relevantes para a área de ET, bem como permitir a replicação deste experimento em outros contextos de *Agile ST*. Para compreensão das fases de desenvolvimento deste experimento, a Figura D.1 ilustra as atividades desenvolvidas, desde o planejamento à avaliação.

D.2 Planejamento do Experimento

As metas deste experimento foram definidas seguindo as orientações do paradigma *Goal Question Metric (GQM)*. Assim, buscamos **analisar** a aplicação da abordagem SBTM, **com a finalidade de** perceber suas contribuições e limitações, **com respeito a** implementação de Testes Exploratórios, **do ponto de vista** de profissionais que integram equipes de desenvolvimento de *software*, **no contexto de** práticas ágeis.

Para alcançar esse objetivo e conduzir essa pesquisa foram definidas as seguintes Questões de Pesquisa (QPs) específicas:

QP_1 : Quais as contribuições de adotar práticas de SBTM no cotidiano da equipe?

QP_2 : Quais as limitações de adotar práticas de SBTM no cotidiano da equipe?

Desse modo, a QP_1 tem como objetivo identificar as principais contribuições da adoção de práticas de SBTM, no que se refere à condução e execução de uma sessão de ET. Enquanto a QP_2 busca investigar as principais limitações percebidas na implementação do SBTM, em relação à execução de uma sessão de ET, elaboração de artefatos de teste, dentre outros aspectos.

Assim, para responder cada QP específica foi planejado e executado um curso com equipes ágeis de desenvolvimento de *software*, para apresentar orientações sobre a aplicação de ET por meio da abordagem SBTM (ver Seção D.2.1).

Após a execução do curso foi feito um acompanhamento das atividades de ET executadas pelos participantes, durante o cotidiano de trabalho com desenvolvimento ágil. Com base nas orientações aprendidas no curso, as sessões de ET foram planejadas e executadas. Em seguida, um *brainstorming* foi conduzido com os profissionais para perceber os benefícios e as limitações vivenciadas neste contexto. A seção D.2.2 descreve o planejamento do *brainstorming*.

D.2.1 Planejamento do Curso de ET

Como demonstra a Figura D.1, a fase de planejamento do curso de ET, consistiu em quatro etapas bem definidas e descritas a seguir.

Etapa 1. Definir o Plano do Curso. Nesta etapa foram definidas a ementa do curso, a carga horária a ser ministrada, a data de realização do curso, o público-alvo, os objetivos a serem alcançados e os materiais de aulas a serem produzidos, de maneira detalhada de acordo com a metodologia adotada. Enfatiza-se que a definição deste Plano de Curso foi amplamente discutido, revisado e avaliado por dois especialistas na área de ST. De forma complementar, foram definidas também as ferramentas a serem usadas no curso, considerando o contexto do ensino remoto - pois o curso foi aplicado em um cenário não-presencial. Assim, foram selecionadas as ferramentas: *Google Meet*, para a comunicação por vídeo durante as aulas; *Discord*, para a comunicação entre os participantes durante as atividades práticas; *Google Drive*, para armazenamento e compartilhamento dos materiais e recursos de aula (em documentos, planilhas e apresentações); *Google Forms*, para elaboração e dis-

ponibilização do questionário de avaliação, após o curso; e, o *Xray Exploratory App*¹, para planejamento e execução dos ET, apenas na última atividade prática.

Etapa 2. Elaborar os materiais de aula. As aulas deste curso tem a finalidade de treinar os participantes sobre o assunto de ET no contexto ágil, especificamente sobre SBTM, e equilibrar o nível de conhecimento entre todos os profissionais ágeis, participantes do curso. Neste contexto, o conteúdo abordado nos materiais de aula foram embasados em [6; 26; 36; 40; 77] e em palestras atuais, conduzidas por profissionais especialistas renomados na área de ET. É importante destacar que foram elaborados (1) notas de aulas (*slides*) com conteúdo teórico e exemplos práticos sobre ET e SBTM, e (2) uma apostila com a síntese detalhada do conteúdo abordado no curso; bem como, apresentada (3) uma lista de ferramentas que apoiam o planejamento e a execução de ET (gerada e exposta na Seção ??), (4) uma lista de vídeos (tutoriais e palestras) disponíveis na *web*, e (5) uma lista de artigos técnicos e livros sobre ET no contexto ágil.

Etapa 3. Elaborar as atividades práticas. Para exercitar e reforçar o aprendizado sobre o conteúdo ministrado em cada módulo do curso foram elaboradas exemplos e atividades práticas, com base nas orientações fornecidas pelas metodologias ativas PBL e JiTT [34; 59]. Nesta etapa foram definidos e elaborados os materiais e recursos necessários para a realização dessas atividades, por exemplo: a seleção do sistema *web* a ser testado; um guia com orientações básicas para cada atividade prática; *templates* dos artefatos de teste (como, *Charters*, *Test Points* e *Session Report*) para otimizar o tempo dedicado à cada atividade; artefatos de requisitos (como, um Documento de Especificação de Requisitos do Sistema e um Diagrama *Use Case*); e, a seleção de um manual de instalação do programa *Xray Exploratory App*.

Etapa 4. Elaborar o questionário de avaliação. Com a finalidade de coletar informações sobre a experiência e o aprendizado dos participantes foi elaborado um questionário *online*², com questões objetivas e subjetivas. Ao total foram incluídas 41 (quarenta e uma) questões, distribuídas entre 39 (trinta e nove) questões de múltipla escolha e 02 (duas) questões abertas,

¹*Xray Exploratory App*: <https://www.getxray.app/exploratory-testing-app/>

²Acesso ao questionário de avaliação: <https://bitly.com/EBCnsw>.

cuja resposta foi opcional ao participante.

O questionário foi projetado no *Google Forms* e organizado em quatro seções. De modo que, a primeira seção objetivou caracterizar, resumidamente, o perfil profissional dos respondentes. A segunda seção, buscou identificar os procedimentos e as práticas organizacionais em relação à prática de ST nas *Sprints* dos projetos desenvolvidos pelos equipes ágeis, antes da oferta do curso. A terceira seção, buscou identificar as percepções do respondente acerca do ensino-aprendizagem obtido durante o curso de ET. E, por fim, a quarta seção objetivou identificar as contribuições do PBL e do JiTT na condução do curso de ET.

É importante destacar que para responder o questionário, os participantes deveriam: (1) ter participado de todos os módulos do curso; (2) ter realizado as atividades práticas desenvolvidas em cada módulo; e, (3) logo no início do questionário, concordar com um Termo de Consentimento Livre e Esclarecido (TCLE) da pesquisa.

Na Tabela 4.6 a estrutura do curso é apresentada, juntamente com a descrição dos tópicos e conteúdos abordados na ementa, e das atividades práticas planejadas para o final de cada módulo de aula. Adicionalmente, é informada a carga horária definida para cada módulo do curso³.

D.2.2 Planejamento do *Brainstorming*

O *Brainstorming* consiste numa técnica utilizada em grupo para geração de ideias inovadoras ou para *insights* sobre uma determinada temática [15]. No geral, um *brainstorming* deve (i) gerar o maior número de ideias possíveis, (ii) estender a interpretação das ideias, (iii) apresentar ideias originais e (iv) realizar a combinação e o melhoramento de ideias existentes.

Para conduzir o *brainstorming* foram definidos uma pergunta principal, ou seja, um problema a ser solucionado, e um conjunto de atividades a serem seguidas. Assim, foi definida a seguinte pergunta: “*O que fazer para conseguir integrar Testes Exploratórios, como prática de teste, no cotidiano do equipe?*”. A partir dessa pergunta principal, outras perguntas específicas foram apresentadas para guiar e contribuir com a geração de ideias (ver Tabela D.1).

³O material deste curso pode ser acessado em: <https://url.gratis/oKDzKc>.

Perguntas específicas

1. Em qué a estratégia de Teste baseada em Sessão atrapalhou a execução do TE ?
 2. Algo impede que os TEs sejam praticados rotineiramente pelo equipe? O que impede, de fato? (processo, ferramenta, equipe, projeto, tempo, etc.)
 3. O que podemos fazer para melhorar a execução dos TEs?
 4. Qual estratégia ou artefato de requisitos é mais útil para auxiliar na realização dos TEs?
 5. O que pode ser feito para tornar esses requisitos mais claros à equipe?
 6. Quais informações são importantes registrar/planejar antes da execução do TE, além das que foram indicadas?
 7. Quais informações são importantes registrar durante a execução do TE, além das que foram indicadas?
 8. Quais informações são importantes registrar após da execução do TE, além das que foram indicadas?
 9. Que práticas foram mais interessantes?
 10. Que práticas não acharam interessante?
 11. Quais benefícios para o dia-a-dia da equipe puderam ser observados no curso?
 12. Diante do aprendizado, o que foi mais difícil de implantar no dia-a-dia?
 13. Algo mudou na prática de teste, da equipe, após o curso? O que mudou?
 14. O que você vislumbra que mudaria na prática de teste, após o curso?
 15. O curso de TE influenciou na incorporação das práticas de teste? O que influenciou, de fato?
 16. TE é útil como prática de teste no contexto de trabalho remoto? O que poderia ser incorporado para contribuir com o trabalho remoto?
-

Tabela D.1: Perguntas específicas do *brainstorming*.

D.3 Execução do Experimento

D.3.1 Execução do Curso

A população deste estudo abrangeu doze profissionais da indústria de desenvolvimento de *software* que atuam com métodos ágeis, em uma mesma organização. No período em que o curso foi aplicado, os profissionais trabalhavam em locais distribuídos geograficamente, em virtude da pandemia do *COrona Virus Disease* (Doença do Coronavírus) ou *Sars-CoV-2*, que acomete a população brasileira, especificamente, desde fevereiro de 2020. Por este motivo, também, o curso de ET foi conduzido em um contexto de ensino completamente remoto.

De um modo geral, a execução do experimento ocorreu como um treinamento interno com as equipes ágeis da organização conforme o planejado, em: quatro encontros virtuais, com a duração de 02 horas cada encontro, nas datas de 06, 07, 12 e 13 de abril de 2021. É importante destacar que o Módulo II foi dividido em dois encontros, devido a extensão do conteúdo ministrado.

Em cada encontro, o conteúdo foi ministrado e os participantes puderam fazer questionamentos e solucionar suas dúvidas durante toda a aula. Na sequência, para exemplificar a teoria discutida, foi feita uma demonstração com exemplos reais. E, em seguida, os participantes foram orientados a exercitar os conhecimentos obtidos por meio de uma atividade prática baseada em um sistema *web* real. Para isso, algumas orientações sobre a atividade foram fornecidas.

Os participantes foram distribuídos em equipes e estimulados a interagir e colaborar entre si, por meio da dinâmica de cada atividade. A resolução de um problema real também buscou estimular nos participantes a investigação, a reflexão e o desenvolvimento de ET relevantes para o contexto analisado na atividade. Essa estratégia foi baseada nas orientações fornecida pelo PBL.

Ao final de cada encontro, os materiais e recursos de aula foram disponibilizados aos participantes para que tivessem um conhecimento prévio do próximo conteúdo a ser discutido no curso. Essa estratégia, baseada no JiTT, buscou estimular a interação entre a ministrante e os participantes do curso, além de viabilizar discussões mais aprofundadas durante a aula e antecipar o *feedback* sobre os materiais e os recursos adotados para o próximo encontro.

Ao final do curso, os participantes foram orientados a preencher um questionário *online*,

cujo propósito foi o de coletar informações sobre a experiência e o aprendizado sobre ET por meio de práticas do PBL e do JiTT.

D.3.2 Execução do *Brainstorming*

Após a execução do curso de ET, os participantes foram motivados a aplicar ET nos projetos que desenvolvem. No total, foram realizadas nove sessões de ET, divididas em dois momentos específicos do projeto. Em seguida, o *brainstorming* foi realizado.

O *brainstorming* ocorreu em um contexto completamente remoto, com os participantes distribuídos localmente. Por este motivo foi adotada a ferramenta *online* Lucidspark⁴ para facilitar a transcrição das ideias e a colaboração entre os participantes distantes fisicamente.

Em relação ao conjunto de atividades seguidas no *brainstorming*, foram planejadas e executadas as seguintes:

1. *Atividade 1. Brainstorming em silêncio.* Esta atividade consiste em gerar ideias, de forma individual, para tentar resolver a problemática apresentada. Assim, os participantes devem escrever suas ideias em Post its. Indica-se que a duração, em média, seja de 15 minutos.
2. *Atividade 2. Compartilhamento das Ideias.* Esta atividade consiste em apresentar as ideias que foram geradas e transcritas nos Post its. É permitido que os outros participantes façam perguntas ou acrescentem alguma nova informação ou ideia. Indica-se que a duração, em média, seja de 20 minutos.
3. *Atividade 3. Filtro de ideias.* O objetivo desta atividade é descartar ideias que não estejam alinhadas com o contexto do problema ou que gerem desentimentos. Indica-se que a duração, em média, seja de 10 minutos.
4. *Atividade 4. Primeira votação.* Nesta atividade, todos os participantes devem selecionar as ideias que melhor solucionam o problema exposto. Apenas as 3 ideias mais votadas são listadas para a próxima atividade. Indica-se que a duração, em média, seja de 5 minutos.

⁴Lucidspark: <https://lucidspark.com/pt>

5. *Atividade 5. Melhoria das Ideias.* O objetivo desta atividade é melhorar as ideias mais votadas, adicionando novas informações importantes, por meio de mais *post its*, com detalhes de artefatos, atividades de testes, documentação, plataformas ou ferramentas de TE, organização da equipe, entre outros. Indica-se que a duração, em média, seja de 10 minutos.
6. *Atividade 6. Segunda votação.* Por fim, os participantes votam, pela segunda vez, na ideia mais aplicável para resolver o problema apresentado. Duração: 4 minutos.

D.4 Procedimentos de Análise

Após a coleta de dados, por meio do questionário *online*, foram gerados relatórios individuais, de acordo com o objetivo de cada seção investigada no questionário. É válido ressaltar que as informações destes relatórios foram anonimadas para preservar a identidade dos participantes.

Assim, para analisar os dados extraídos dos conteúdos das respostas fornecidas pelos participantes foi conduzida uma análise quantitativa [79], principalmente nas respostas fornecidas por meio da Escala *Likert*, com opções de 1 a 5 (sendo: 1 - Discordo Totalmente; 2 - Discordo Parcialmente; 3 - Nem concordo, nem discordo; 4 - Concordo Parcialmente; 5 - Concordo Totalmente). Neste sentido, as respostas foram analisadas por classe: discordância, indecisão e concordância. De forma adicional, uma análise qualitativa foi conduzida nas respostas das questões subjetivas, mas como se tratavam de respostas opcionais ou complementares às questões objetivas foram poucas as necessidades de aplicação deste tipo de análise.

Em relação ao *brainstorming* também foi aplicada análise qualitativa [79] nas respostas fornecidas, após a atividade de “Melhoria das Ideias”.

Apêndice E

Apêndice E - Questionário aplicado na *Validação da Solução no Ciclo 1*

I - Dados do Participante

1. Nome e Sobrenome.
2. Email.
3. Grau de instrução acadêmica.
 - Graduação em andamento.
 - Graduação completa.
 - Especialização em andamento.
 - Especialização completa.
 - Mestrado em andamento.
 - Mestrado completo.
 - Doutorado em andamento.
 - Doutorado completo.
4. Quantos anos de experiência você possui na área de Teste de Software.
 - Até 1 ano.
 - Entre 1 e 2 anos.
 - Entre 2 e 3 anos.
 - Entre 3 e 4 anos.
 - Entre 4 e 5 anos.
 - Mais de 5 anos.

5. Conte a sua experiência com Teste de Software (ou, mais especificamente, com Testes Exploratórios).

6. Se você participou de projetos que envolvem ou já aplicou Testes Exploratórios nos conte sua experiência.

II - Facilidade de entendimento do Método Agile ETeasy

(Nas questões 07 à 10, analise as afirmativas e informe sua opinião de acordo com a escala de concordância: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Nem discordo e nem concordo; (4) Concordo parcialmente; (5) Concordo totalmente.)

7. Foi fácil entender as etapas, atividades e tarefas do método?

8. Entendi claramente como utilizar as técnicas sugeridas pelo método.

9. Como podemos melhorar a descrição das tarefas para facilitar o entendimento das mesmas?

10. Entendi claramente como preencher o artefato de teste e o relatório do método.

11. Se você tiver outro comentário, crítica ou sugestão de melhoria que não foram contempladas em forma de questionamento nas questões anteriores, informe aqui.

III - Utilidade percebida

(Nas questões 12 à 19, analise as afirmativas e informe sua opinião de acordo com a escala de concordância: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Nem discordo e nem concordo; (4) Concordo parcialmente; (5) Concordo totalmente.)

12. Considero adequadas as etapas definidas no método Agile ETeasy para realizar Testes Exploratórios.

13. Considero adequadas as atividades definidas no método Agile ETeasy para realizar os Testes Exploratórios.

14. Considero adequadas as tarefas definidas no método Agile ETeasy para realizar os Testes Exploratórios.

15. Considero adequada a técnica de *Brainwriting* na etapa de Planejamento.

16. Considero adequada a técnica de Pesquisa Documental na etapa de Planejamento.

17. Considero adequada a técnica de *Checklist* de Heurísticas na atividade de “Aplicar os Testes” da etapa de Execução.

18. Considero adequada a técnica de *Workshop* na atividade de “Alinhar as melhorias” da etapa de Encerramento.

19. Considero adequada a técnica de *Brainstorming* na atividade de “Alinhar as melhorias” da etapa de Encerramento.

20. Você tem sugestões de outras atividades a serem alteradas ou adicionadas nas etapas do método, de modo a torná-lo mais útil na execução dos Testes Exploratórios. Qual (is) e em qual (is) etapa (s)?

21. Você tem sugestões de outras tarefas a serem alteradas ou adicionadas nas atividades do método, de modo a torná-lo mais útil na execução dos Testes Exploratórios. Qual (is) e em qual (is) atividade (s)?

22. Você tem sugestões de outras técnicas que podem contribuir com a execução das atividades do método?

23. Com base nas respostas anteriores (14-20), você sugere a inclusão ou remoção de alguma técnica citada? Se sim, qual?

24. Como foi a sua experiência geral durante a leitura do método?

25. Como podemos melhorar o método? O que sentiu falta e gostaria que fosse implementado no método?

26. Se você tiver outro comentário, crítica ou sugestão de melhoria que não foram contempladas em forma de questionamento nas questões anteriores, informe aqui.

Apêndice F

Apêndice F - Questionário aplicado na *Validação da Solução no Ciclo 2*

I - Dados do Participante

01. Nome e Sobrenome (opcional).

02. Email (opcional).

03. Grau de instrução acadêmica.

Graduação em andamento.

Graduação completa.

Especialização em andamento.

Especialização completa.

Mestrado em andamento.

Mestrado completo.

Doutorado em andamento.

Doutorado completo.

04. Qual o tempo de experiência profissional você possui com métodos ágeis?

Até 1 ano.

Entre 1 e 2 anos.

Entre 2 e 3 anos.

Entre 3 e 4 anos.

Entre 4 e 5 anos.

Mais de 5 anos.

05. No contexto ágil, por vezes, o profissional precisa atuar em diferentes funções na equipe. Conte-nos a sua experiência com Teste de Software (se você costuma aplicar teste nas aplicações que desenvolve, como realiza essa tarefa, qual teste aplica, dentre outras características que você considera importante citar).

06. Se você participou de projetos que envolvem, ou já aplicou, Testes Exploratórios nos conte sua experiência.

II - Facilidade de Entendimento do Método Agile ETeasy

(Nas questões 7 à 10, analise as afirmativas e informe sua opinião de acordo com a escala de concordância: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Nem discordo e nem concordo; (4) Concordo parcialmente; (5) Concordo totalmente.)

07. Foi fácil entender as etapas, atividades e tarefas do método.

08. Entendi claramente como utilizar as técnicas (*Brainwriting*, Pesquisa documental, *Pair review*, *Workshop*, *Braisntorming*) sugeridas pelo método.

09. A descrição das tarefas estavam claras.

10. Entendi claramente como preencher o artefato de planejamento do teste e o relatório de encerramento.

11. Você tem sugestões para facilitar o entendimento das ETAPAS, ATIVIDADES e TAREFAS do método? Como podemos melhorá-las? (Se possível, destaque aspectos de melhoria para facilitar a aplicação do método por outros usuários)

12. Você tem sugestões para facilitar o entendimento do ARTEFATO DE PLANEJAMENTO DO TESTE e do RELATÓRIO DE ENCERRAMENTO? Como podemos melhorá-los? (Se possível, destaque aspectos de melhoria para facilitar a aplicação do método por outros usuários)

13. Se você tiver outro comentário, crítica ou sugestão de melhoria que não foram contempladas em forma de questionamento nas questões anteriores, informe aqui.

III - Utilidade do Método Agile ETeasy

(Nas questões 14 à 21, analise as afirmativas e informe sua opinião de acordo com a escala de concordância: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Nem discordo e nem concordo; (4) Concordo parcialmente; (5) Concordo totalmente.)

14. Considero adequadas as ETAPAS definidas no método *Agile ETeasy* para realizar Testes Exploratórios.

15. Considero adequadas as ATIVIDADES definidas no processo Agile ETeasy para realizar os Testes Exploratórios.

16. Considero adequadas as TAREFAS definidas no método *Agile ETeasy* para realizar os Testes Exploratórios.

17. Considero adequada a TÉCNICA DE *BRAINWRITING* na etapa de Planejamento.

18. Considero adequada a TÉCNICA DE PESQUISA DOCUMENTAL na etapa de Planejamento.

19. Considero adequada a TÉCNICA DE *CHECHKLIST* DE HEURÍSTICAS na atividade de “Aplicar os Testes” da etapa de Realização.

20. Considero adequada a TÉCNICA DE *WORKSHOP* na atividade de “Alinhar as melhorias” da etapa de Encerramento.

21. Considero adequada a TÉCNICA DE *BRAINSTORMING* na atividade de “Alinhar as melhorias” da etapa de Encerramento.

22. Você tem sugestões de outras ATIVIDADES, TAREFAS ou TÉCNICAS a serem alteradas ou adicionadas nas etapas do método à torná-lo mais útil na execução dos Testes Exploratórios. Qual (is) e em qual (is) etapa (s)?

23. Com base nas respostas anteriores (17-21), você sugere a inclusão ou remoção de alguma técnica citada? Se sim, qual?

24. Como foi a sua experiência geral durante a leitura do processo, por meio do *website* (<https://sites.google.com/splab.ufcg.edu.br/processoagileeteasy/agile-eteasy>)?

25. Se você tiver outro comentário, crítica ou sugestão de melhoria que não foram contempladas em forma de questionamento nas questões anteriores, informe aqui.

Apêndice G

Apêndice G - Protocolo do Estudo de Caso da etapa de *Implementação da Solução do Ciclo 1 e 2*

Neste Apêndice é apresentado o protocolo adotado na realização do Estudo de Caso aplicado na etapa de *Implementação da Solução do Ciclo 1 e 2*. O Estudo de Caso referiu-se à implementação do método *Agile ETeasy* em um projeto de desenvolvimento de *software* desenvolvido por profissionais que integram equipes ágeis. Uma contextualização é apresentada na Seção G.1. Em seguida, é exposto o *design* do Estudo de Caso na Seção G.2. Depois disso, é descrita a preparação para a coleta de dados na Seção G.3, seguida da descrição da coleta de dados, na Seção G.4. E, por fim, são discutidos os procedimentos de análise na Seção G.5.

G.1 Visão Geral

A metodologia de estudo de caso é adequada para muitos tipos de pesquisa em Engenharia de *Software*, principalmente quando os objetos de estudo são fenômenos contemporâneos, difíceis de estudar isoladamente [79]. Os estudos de caso fornecem uma compreensão mais profunda dos fenômenos estudados em seu contexto real, complementa Wohlin et al. [79].

Por isso, na etapa de *Implementação da Solução do Ciclo 1 e 2* da DSR, foi adotado, como instrumento de pesquisa, a realização de um Estudo de Caso. Para Wohlin et al. [79],

ao realizar um estudo de caso, há cinco etapas principais que devem ser seguidas: (1) *Design* do estudo de caso; (2) Preparação para coleta de dados; (3) Coleta de dados; (4) Análise dos dados coletados; e, (5) Relatórios. Nas seções seguintes detalhamos o planejamento do Estudo de Caso 1 e 2, aplicados na DSR.

G.2 Design do Estudo de Caso

As metas deste estudo de caso foram definidas seguindo as orientações do paradigma *Goal Question Metric* (GQM) [79]. Assim, buscamos **analisar** a implementação do método *Agile ETeasy*, **com a finalidade de** avaliar sua facilidade de entendimento e de uso, **com respeito a** inserção de práticas de Testes Exploratórios, **do ponto de vista** de profissionais que integram equipes de desenvolvimento de *software*, **no contexto de** métodos ágeis.

Para alcançar esse objetivo e conduzir esse estudo foram definidas as seguintes Questões de Pesquisa (QPs) específicas:

QP_1 : Como garantir a facilidade de entendimento do método *Agile ETeasy*, de modo a torná-lo aplicável ao cotidiano de equipes ágeis?

QP_2 : Como garantir a facilidade de uso do método *Agile ETeasy*, de modo a torná-lo aplicável ao cotidiano de equipes ágeis?

Desse modo, a QP_1 tem como objetivo identificar a facilidade de entendimento do método *Agile ETeasy* por meio da avaliação dos seus elementos, por profissionais que integram equipes ágeis. Enquanto que a QP_2 busca investigar a facilidade de uso do método *Agile ETeasy* por meio da avaliação dos seus elementos, também, por profissionais que integram equipes ágeis. Para responder essas questões de pesquisa, o método *Agile ETeasy* deve ser aplicado de forma teórica e prática, em um contexto de um projeto real.

G.3 Preparação para coleta de dados

Para a coleta de dados, foram elaborados um questionário *online* (conforme apresentado no Apêndice H), com o objetivo de coletar informações sobre a caracterização dos participantes e sobre sua experiência com Teste de *Software*. No total, foram abordadas cinco questões subjetivas e três questões objetivas. Em seguida, foi conduzido um minicurso explicativo do

método *Agile ETeasy* para informar detalhes do método aos participantes. Depois disso, os participantes foram orientados a aplicar o método em seus projetos.

Esse estudo de caso também abrangeu a etapa de *Avaliação da Implementação* do Ciclo 1 e 2 da DSR aplicada nesta tese. Desse modo, a avaliação ocorreu por meio de um Diário de Bordo *online* (conforme apresentado no Apêndice I), para detalhamento da experiência do participante na utilização dos elementos do método *Agile ETeasy*, pontos positivos, resultados alcançados, pontos negativos e/ou dificuldades encontradas. Outro instrumento de pesquisa adotado foi a aplicação de entrevistas (conforme apresentado no Apêndice J) com cada participante para obter informações que não ficaram claras ou que não foram possíveis de ser transcritas no Diário de Bordo.

G.4 Coleta de Dados

No Estudo de Caso 01, o método *Agile ETeasy* foi implementado com uma equipe ágil composta por três profissionais. No Estudo de Caso 02, o método foi implementado com uma equipe ágil composta por quatro profissionais. Em ambos os estudos, os participantes foram estimulados preencher um questionário *online* inicial para caracterização do seu perfil profissional. Em seguida, um mini-curso do método *Agile ETeasy* foi conduzido para apresentar os elementos do método e embasar seu uso. Depois disso, os participantes foram organizados em pares para aplicar o método *Agile ETeasy* em funcionalidades de um projeto real desenvolvido pelo equipe. Em paralelo ao uso do método, os participantes poderiam preencher o Diário de Bordo com suas impressões ou dúvidas iniciais. Após a finalização do ET, por meio do método *Agile ETeasy*, os participantes foram entrevistados.

É importante destacar que ambos os Estudos de Caso foram aplicados durante a pandemia do *COrona VIRus Disease* (Doença do Coronavírus) ou *Sars-CoV-2*, que acomete a população brasileira, especificamente, desde fevereiro de 2020. E, por esse motivo, os profissionais participaram dos estudos remotamente.

O Estudo de Caso 01 ocorreu em um encontro virtual, com duração de 03 horas, na data de 14 de abril de 2022. O Estudo de Caso 02 ocorreu em dois encontros virtuais, com duração de 01 hora e 30 minutos (cada encontro) na data de 30 de junho de 2022.

G.5 Análise dos dados coletados

Após a coleta de dados, por meio do questionário *online*, do Diário de Bordo e das entrevistas, foram gerados relatórios individuais, de acordo com o objetivo de cada instrumento de pesquisa adotado. É válido ressaltar que as informações destes relatórios foram anonimadas para preservar a identidade dos participantes.

Em síntese, os dados gerados foram analisados e relatados de forma qualitativa. Para isso, analisamos o material textual coletado, desde os produtos de mídia gerados nas entrevistas às informações fornecidas no questionário - nas respostas às questões subjetivas ou nas respostas complementares às questões objetivas - e, no diário de bordo.

Uma análise quantitativa [79] também foi aplicada nas respostas às questões objetivas contidas no questionário *online*, principalmente nas respostas fornecidas por meio da Escala *Likert*, com opções de 1 a 5 (sendo: 1 - Discordo Totalmente; 2 - Discordo Parcialmente; 3 - Nem concordo, nem discordo; 4 - Concordo Parcialmente; 5 - Concordo Totalmente). Neste sentido, as respostas foram analisadas por classe: discordância, indecisão e concordância.

Apêndice H

Apêndice H - Questionário inicial da *Implementação da Solução do Ciclo 1 e 2*

I - Dados do Participante

- 01. Nome e Sobrenome.
- 02. Email para contato.
- 03. Grau de instrução acadêmica.
 - () Graduação em andamento.
 - () Graduação completa.
 - () Especialização em andamento.
 - () Especialização completa.
 - () Mestrado em andamento.
 - () Mestrado completo.
 - () Doutorado em andamento.
 - () Doutorado completo.

II - Experiência com Teste de Software

- 04. Quantos anos de experiência você possui na área de Teste de Software.
 - () Até 1 ano.
 - () Entre 1 e 2 anos.
 - () Entre 2 e 3 anos.
 - () Entre 3 e 4 anos.
 - () Entre 4 e 5 anos.

Mais de 5 anos.

05. Conte a sua experiência com Teste de Software.

06. Você costuma testar as funcionalidades/aplicações que implementa?

Sim, com frequência.

Sim, esporadicamente.

Não, pois outro profissional é quem realiza os testes.

Não, porque não gosto.

Não, porque não tenho conhecimento da área.

Outro:

06.1 Se sua resposta foi “Sim” na Questão 6: como você costuma realizar os testes (técnica usada, periodicidade do teste, etc.)?

07. Conte sua experiência com Testes Exploratórios.

08. Se você participou de projetos que envolvem ou já aplicou Testes Exploratórios nos conte sua experiência.

Apêndice I

Apêndice I - Diário de Bordo usado na *Avaliação da Implementação do Ciclo 1 e 2*

O objetivo deste Diário de Bordo é detalhar como foi a experiência de implementação do Método *Agile ETeasy* em um projeto real de uma equipe ágil, conforme as etapas, as atividades, as tarefas, as técnicas e os artefatos utilizados. Se houver algum comentário, sugestão ou dificuldade encontrada que não esteja relacionada diretamente a algum dos elementos do referido método, deixe em branco os demais campos e escreva, apenas, no campo definido como “Reflexão Crítica”. O campo “Reflexão Crítica” é destinado ao relato sobre sua experiência na utilização dos elementos do Método *Agile ETeasy*, pontos positivos, resultados alcançados, pontos negativos e/ou dificuldades encontradas.

Tabela I.1: Registro da experiência de implementação do Método *Agile ETeasy*.

| Data | Etapas | Atividade | Tarefa | Técnica | Artefato | Reflexão Crítica |
|------|--------|-----------|--------|---------|----------|------------------|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Apêndice J

Apêndice J - Entrevista realizada na *Avaliação da Implementação do Ciclo 1 e* **2**

I - Sobre a Facilidade de Entendimento do Método *Agile ETeasy*

01. Você achou fácil entender as etapas, atividades e tarefas do método?
02. Você gostaria de dar alguma sugestão para facilitar o entendimento das etapas, atividades e tarefas do método?
03. Você conseguiu compreender como utilizar as técnicas (*Brainwriting*, Pesquisa documental, *Pair review*, *Workshop*, *Braisntorming*) sugeridas pelo método?
04. A descrição das tarefas estavam claras?
05. Você conseguiu compreender como preencher o artefato de teste (planejamento)? E o relatório (de encerramento)? Tem alguma sugestão de melhoria?

II - Sobre a Utilidade do Método *Agile ETeasy*

06. Qual sua opinião sobre a utilidade das ETAPAS, ATIVIDADES e TAREFAS definidas no método *Agile ETeasy*, para realizar os Testes Exploratórios?
07. Qual sua opinião sobre a utilidade das técnicas indicadas no método *Agile ETeasy*? (*Brainwriting* e Pesquisa Documental, no Planejamento; *Checklist* de Heurísticas, na Realização; e, *Workshop* e *Brainstorming*, no Encerramento)
08. Você tem sugestões de outras atividades a serem alteradas, adicionadas ou removidas do método *Agile ETeasy*, de modo à torná-lo mais útil na execução dos Testes Exploratórios.

Qual (is) e em qual (is) etapa (s)?

09. Como foi a sua experiência geral durante a leitura do método?
10. Como podemos melhorar o método *Agile ETeasy*? O que você sentiu falta e gostaria que fosse implementado no método?
11. Quais das técnicas indicadas no método foram, de fato, utilizadas? (*Brainwriting*, Pesquisa documental, *Pair review*, *Workshop*, *Braisntorming*)
12. Qual sua experiência com uso desta técnica (citada)?
13. Você tem outro comentário, crítica ou sugestão de melhoria que não foram mencionadas aqui?

Apêndice K

Apêndice K - Método *Agile ETeasy*:

Artefato de Planejamento

| Planejamento do TE | |
|--------------------|---|
| Data: | <dd/mm/aaaa> |
| Sistema: | <identificação do nome do sistema/projeto> |
| Versão: | X.X.X |
| Sprint: | <caracterização da <i>Sprint</i> (nome, numeração ou referência)> |
| Testadores: | <identificação do profissional (is) que realizará (ão) o teste> |

| ID | Referência ao Requisito | Cenário de teste | Testadores |
|----|---|-------------------------------------|--|
| 01 | <Referência à <i>Use Case</i> , <i>User Stories</i> , documento, ou Sistema de Gerenciamento de Requisitos> | <Descrição do escopo a ser testado> | <Identificação do profissional (is) responsável (is) por testar o referido escopo> |
| 02 | | | |
| 03 | | | |
| 04 | | | |
| 05 | | | |
| 06 | | | |
| 07 | | | |
| 08 | | | |
| 09 | | | |
| 10 | | | |
| 11 | | | |
| 12 | | | |
| 13 | | | |
| 14 | | | |
| 15 | | | |

Apêndice L

Apêndice L - Método *Agile ETeasy*: Artefato de Encerramento

| Relatório de Encerramento do TE | |
|---------------------------------|---|
| Testadores | <identificação do profissional (is) responsável (is) por testar o referido escopo> Duração: xx:xx minutos |
| Escopo testado | <Descrição do escopo a ser testado - assim como está no Planejamento> |
| Arquivo de dados | Usuários: <Informe o nível de acesso do usuário na funcionalidade testada> |
| | Configuração: <Informe a configuração do sistema, em que o teste foi executado, se houver. Ou informações de configuração de hardware que são importantes de serem informadas para entendimento do teste. > |
| | Arquivos: <Informe os arquivos necessários para execução da funcionalidade testada, como arquivos de carga, de entrada/saída de dados, dentre outros referentes ao contexto da aplicação, e se houver.> |
| | Outros: <Outras informações importantes a serem registradas, como: dispositivos de hardware/equipamentos usados no teste, dentre outros> |
| Defeitos | 1. <Informe a descrição dos defeitos encontrados, se houver.> 2. 3. |
| Melhorias | <Informe sugestões de melhorias a serem implementadas a partir dos defeitos percebidos, se houver.> 1. 2. 3. |
| Pendências | <Informe as funcionalidades que não foram finalizadas, ou seja, pendências percebidas durante o teste, se houver.> 1. 2. 3. |
| Notas | <Informe notas adicionais sobre o teste executado, se houver. Informações importantes para o time.> 1. 2. 3. |
| Anexos | <Informe os arquivos (documentos ou links) adicionais, necessários à execução ou compreensão da funcionalidade testada, ou de acesso à registros do teste, como: 1. gifs ou prints de tela dos bugs encontrados> 2. 3. |