



**INSTITUTO
FEDERAL**
Paraíba

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus João Pessoa

Programa de Pós-Graduação em Tecnologia da Informação

Nível Mestrado Profissional

THIAGO AUGUSTO FIRMINO BARBOSA

**ESPECIFICAÇÃO DE REQUISITOS EM PROJETOS ÁGEIS:
AVALIAÇÃO DE UMA ABORDAGEM VOLTADA PARA
EQUIPES DE DESENVOLVIMENTO**

DISSERTAÇÃO DE MESTRADO

JOÃO PESSOA - PB

2024

Thiago Augusto Firmino Barbosa

**ESPECIFICAÇÃO DE REQUISITOS EM PROJETOS ÁGEIS:
AVALIAÇÃO DE UMA ABORDAGEM VOLTADA PARA
EQUIPES DE DESENVOLVIMENTO**

Dissertação de Mestrado apresentada como requisito parcial para obtenção do título de Mestre em Tecnologia da Informação pelo Programa de Pós-Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB.

Orientadora: Prof. Dra. Juliana Dantas Ribeiro Viana
de Medeiros

Coorientadora: Prof. Dra. Heremita Brasileiro Lira

João Pessoa - PB

2024

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Nilo Peçanha - *Campus* João Pessoa, PB.

B238e Barbosa, Thiago Augusto Firmino.

Especificação de requisitos em projetos ágeis : avaliação de uma abordagem voltada para equipes de desenvolvimento / Thiago Augusto Firmino Barbosa. - 2024.

86 f. : il.

Dissertação (Mestrado em Tecnologia da Informação) – Instituto Federal de Educação da Paraíba / Programa de Pós-Graduação em Tecnologia da Informação (PPGTI), 2024.

Orientação : Profa. Dra Juliana Dantas Ribeiro Viana de Medeiros.

Coorientação : Profa. Dra Heremita Brasileiro Lira.

1. *RSD (Requirements specification for developers)*. 2. Especificação de requisitos. 3. Pesquisa-ação. 4. Desenvolvimento ágil de *software*. I. Título.

CDU 001.891:004.41(043)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU*
MESTRADO PROFISSIONAL EM TECNOLOGIA DA INFORMAÇÃO

THIAGO AUGUSTO FIRMINO BARBOSA

**ESPECIFICAÇÃO DE REQUISITOS EM PROJETOS ÁGEIS: AVALIAÇÃO DE UMA ABORDAGEM
VOLTADA PARA EQUIPES DE DESENVOLVIMENTO**

Dissertação apresentada como requisito para obtenção do título de Mestre em Tecnologia da Informação, pelo Programa de Pós- Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB - Campus João Pessoa.

Aprovado em 30 de agosto de 2024

Membros da Banca Examinadora:

Dra. Juliana Dantas Ribeiro Viana de Medeiros

IFPB - PPGTI

Dra. Heremita Brasileiro Lira

IFPB - PPGTI

Dra. Ianna Maria Sodre Ferreira de Sousa

IFPB

Dr. Francisco Petrônio Alencar de Medeiros

IFPB - PPGTI

João Pessoa/2024

Documento assinado eletronicamente por:

- Heremita Brasileiro Lira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 30/08/2024 11:08:31.
- Francisco Petronio Alencar de Medeiros, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/09/2024 22:38:03.
- Juliana Dantas Ribeiro Viana de Medeiros, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 30/08/2024 10:23:00.
- Ianna Maria Sodre Ferreira de Sousa, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 30/08/2024 10:24:59.

Este documento foi emitido pelo SUAP em 15/08/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 592607
Verificador: 89caf1a16b
Código de Autenticação:



Av. Primeiro de Maio, 720, Jaguaribe, JOAO PESSOA / PB, CEP 58015-435
<http://ifpb.edu.br> - (83) 3612-1200

AGRADECIMENTOS

Desejo expressar minha gratidão, em primeiro lugar, a Deus, por ter me dado a capacidade de enfrentar os desafios ao longo do caminho.

À minha querida esposa, Christiane Holanda Arantes Barbosa, por todo o apoio e companheirismo demonstrados.

Aos meus pais, Romildo e Eliane, que dedicaram seus esforços ao meu crescimento.

A todos os meus familiares, que me apoiaram de diversas maneiras, tornando a jornada acadêmica menos difícil.

À minha orientadora, Juliana Dantas, e à coorientadora, Heremita Brasileiro, pelo companheirismo demonstrado durante a execução desta pesquisa.

Por fim, a todos que contribuíram, direta ou indiretamente, para tornar este trabalho uma realidade.

RESUMO

A especificação de requisitos desempenha um papel fundamental no desenvolvimento de software. Entretanto, se realizada de forma inadequada pode acarretar uma diminuição da produtividade das equipes, maior complexidade de manutenção e escassez de informações necessárias para desenvolvimento de um software. A literatura apresenta algumas estratégias para atuar na resolução ou diminuição desses problemas, como exemplo a abordagem RSD (*Requirements Specification for Developers*) que propôs uma Especificação de Requisitos de Software (ERS) para projetos ágeis, escrita em uma linguagem voltada para equipes de desenvolvimento ao invés de orientada ao cliente. No entanto, até o momento, não encontramos na literatura, outros estudos que tenham avaliado a abordagem, apresentando seus pontos positivos e negativos. Nesse contexto, o presente trabalho teve como objetivo avaliar a abordagem RSD em um projeto ágil de uma empresa de software. Para isso, foi conduzida uma pesquisa de natureza aplicada, utilizando uma abordagem qualitativa e exploratória. O procedimento técnico adotado foi do tipo pesquisa-ação, permitindo uma intervenção direta no ambiente de trabalho para analisar e avaliar os resultados da implementação da abordagem. Inicialmente foi feita uma avaliação de como as atividades de requisitos estavam sendo realizadas no projeto. Em seguida, a abordagem RSD foi implantada em um projeto de desenvolvimento, e foi feito um acompanhamento durante algumas sprints para coletar os dados necessários para uma análise sistemática da abordagem. Os resultados obtidos após análise e síntese dos dados coletados apontaram que a abordagem RSD contribuiu para uma melhor definição e compreensão dos requisitos por parte das equipes de desenvolvimento, proporcionando uma base mais sólida para a construção de software. Além disso, a utilização de protótipos e a especificação detalhada dos critérios de aceitação melhoraram a comunicação entre os membros da equipe e facilitaram a identificação de requisitos funcionais, não funcionais e técnicos de forma mais integrada. A implementação da RSD também permitiu um melhor alinhamento entre as expectativas dos stakeholders e a equipe de desenvolvimento, resultando em entregas mais alinhadas às necessidades do projeto. Contudo, a aplicação da abordagem RSD apresentou alguns desafios, incluindo a necessidade de um maior investimento de tempo na fase inicial do projeto para a capacitação da equipe e a adaptação dos processos existentes. Além disso, a complexidade adicional introduzida pela necessidade de manter uma documentação mais detalhada demandou um esforço adicional de gestão por parte da equipe de desenvolvimento. Outra oportunidade de melhoria identificada foi a necessidade de desenvolver ferramentas ou metodologias complementares que auxiliem na manutenção e atualização dos artefatos gerados pela abordagem RSD, de modo a suportar de forma eficaz a dinâmica de mudanças dos requisitos em projetos ágeis.

Palavras-chaves: RSD (*Requirements Specification for Developers*), Especificação de requisitos, Pesquisa-ação, Desenvolvimento Ágil de Software.

ABSTRACT

The specification of requirements plays a fundamental role in software development. However, if performed inadequately, it can lead to decreased team productivity, increased maintenance complexity, and a lack of necessary information for software development. The literature presents some strategies to address or mitigate these problems, such as the RSD (Requirements Specification for Developers) approach, which proposed a Software Requirements Specification (SRS) for agile projects, written in a language oriented towards development teams rather than customers. However, so far, we have not found other studies in the literature that have evaluated the approach, presenting its positive and negative points. In this context, the present work aimed to evaluate the RSD approach in an agile project of a software company. For this, an applied research was conducted, using a qualitative and exploratory approach. The technical procedure adopted was action research, allowing direct intervention in the work environment to analyze and evaluate the results of the implementation of the approach. Initially, an assessment was made of how the requirements activities were being carried out in the project. Subsequently, the RSD approach was implemented in a development project, and monitoring was conducted during several sprints to collect the necessary data for a systematic analysis of the approach. The results obtained after analysis and synthesis of the collected data indicated that the RSD approach contributed to a better definition and understanding of the requirements by the development teams, providing a more solid foundation for software construction. Furthermore, the use of prototypes and the detailed specification of acceptance criteria improved communication among team members and facilitated the identification of functional, non-functional, and technical requirements in a more integrated manner. The implementation of RSD also allowed for better alignment between stakeholders' expectations and the development team, resulting in deliveries that were more aligned with the project's needs. However, the application of the RSD approach presented some challenges, including the need for a greater investment of time in the initial phase of the project for team training and the adaptation of existing processes. Additionally, the added complexity introduced by the need to maintain more detailed documentation required additional management effort from the development team. Another improvement opportunity identified was the need to develop complementary tools or methodologies to assist in the maintenance and updating of artifacts generated by the RSD approach, in order to effectively support the dynamic nature of requirements changes in agile projects.

Keywords: *RSD (Requirements Specification for Developers), Requirement Specification, Action Research, Agile Software Development.*

LISTA DE FIGURAS

Figura 1 - Estrutura para a condução da pesquisa-ação	18
Figura 2 - Estrutura do artefato RSD.....	27
Figura 3 - Processo de síntese temática	40
Figura 4 - Item de backlog do produto	41
Figura 5 - Modelo da abordagem criado no <i>Google Sheets</i>	45
Figura 6 - Campo adicionado ao <i>Azure DevOps</i> com link do RSD	46
Figura 7 - Legendas de cores das abas do modelo.....	48
Figura 8 - Novos campos para dependências, interfaces e responsáveis	49
Figura 9 - Novos campos para informações adicionais, serviços e links úteis.	49
Figura 10 - Percentuais de RSDs avaliados no critério de objetividade.	52
Figura 11 - Percentuais de RSDs avaliados no critério de legibilidade.	53
Figura 12 - Percentuais de RSDs avaliados no critério de completude.....	53
Figura 13 - Percentuais de RSDs avaliados no critério de volatilidade.	54
Figura 14 - Percentuais de RSDs avaliados positivamente e negativamente.....	55

LISTA DE TABELAS

Tabela 1 – Exemplos de critérios de aceitação.	31
Tabela 2 – Indivíduos envolvidos na pesquisa.	37
Tabela 3 – Publicações desta pesquisa.	65

LISTA DE ABREVIATURAS E SIGLAS

AC+	<i>Acceptance Criteria Plus</i>
DAS	Desenvolvimento Ágil de Software
DSDM	<i>Dynamic Systems Method</i>
ERS	Especificação de Requisitos de Software
FDD	<i>Feature Driven Development</i>
ID	Identificação
NL	<i>Natural Language</i>
RSD	<i>Requirements Specification for Developers</i>
SRS	<i>Software Requirements Specification</i>
STD	<i>State Transition Diagram</i>
TDD	<i>Test Driven Development</i>
XP	<i>Extreme Programming</i>

LISTA DE SÍMBOLOS

%	Por cento
#	Cerquilha

SUMÁRIO

1. INTRODUÇÃO	15
1.1. Justificativa e Definição do Problema	16
1.2. Objetivos	16
1.3. Metodologia	17
1.4. Aplicabilidade	19
1.5. Estrutura do Documento	19
2. FUNDAMENTAÇÃO TEÓRICA	21
2.1. Especificação de Requisitos no Desenvolvimento Ágil de Software	21
2.2. Metodologias e Métodos Ágeis	22
2.2.1. Scrum	23
2.2.2. Extreme Programming	24
2.2.3. Feature Driven Development	24
2.3. A Abordagem Requirements Specification for Developers (RSD)	25
2.3.1 Modelagem Conceitual	28
2.3.2 Prototipação da Interface do Usuário	29
2.3.3 Critérios de Aceitação	30
2.4. Trabalhos Relacionados	32
2.5. Considerações Finais	34
3. AVALIAÇÃO NA EMPRESA ANTES DA IMPLANTAÇÃO DA ABORDAGEM RSD	35
3.1. Protocolo de Pesquisa-ação	35
3.2. Iniciação do Projeto de Pesquisa-Ação	36
3.2.1. Definição da Estrutura Conceitual-Teórica	36
3.2.2. Definição da Unidade de Análise	36
3.2.3. Definição do Contexto e Propósito	36
3.2.3.1. Definição do Tema e Interessados	37
3.3. Procedimentos para Coleta, Análise e Síntese dos Dados	38
3.4. Resultados da Avaliação	41
3.4.1. Processo de Especificação Utilizado no Projeto Antes da Implantação da Abordagem RSD	41
3.4.2. Pontos Positivos Identificados	42
3.4.3. Principais Problemas Identificados	42
3.4.4. Oportunidades de Melhoria Identificadas	43
3.5. Considerações Finais	43
4. IMPLANTAÇÃO DA ABORDAGEM RSD	44
4.1. Capacitação dos Envolvidos na Pesquisa e Aplicação da Abordagem	44
4.1.1. Modelo de Capacitação	46
4.1.2. Dificuldades e Lições Aprendidas Durante a Capacitação	47

4.2. Monitoramento da Utilização da Abordagem	47
4.3. Dificuldades Identificadas pelo Pesquisador no Processo de Implantação	48
4.4. Considerações Finais	50
5. AVALIAÇÃO DA ABORDAGEM RSD	51
5.1. Resultados da Aplicação do Roteiro das Entrevistas	51
5.1.1. Conteúdo	51
5.1.2. Estrutura	55
5.2. Comparação da abordagem RSD em relação ao estado anterior	56
5.2.1. Esforço Necessário para Compreender, Codificar e Testar	56
5.2.2. Esforço Necessário para Estruturação	57
5.2.3. Análise de Impacto de Pedidos de Mudança	57
5.2.4. Transferência de Conhecimento, Colaboração do Cliente e Dependência Entre as Partes Interessadas	58
5.3. Resultados da Avaliação dos Critérios de Pesquisa-Ação	58
5.4. Síntese Temática dos Resultados Após Aplicação da Abordagem	59
5.4.1. Processo de Síntese Temática	59
5.5. Ameaças a Validade	61
5.6. Considerações Finais	62
6. CONCLUSÃO	63
6.1. Contribuições da Pesquisa	64
6.2. Trabalhos Futuros	66
REFERÊNCIAS BIBLIOGRÁFICAS	68
APÊNDICE A – ROTEIRO DA ENTREVISTA PARA COLETA DE DADOS ANTES DA INTRODUÇÃO A ABORDAGEM RSD	73
APÊNDICE B – ROTEIRO DA ENTREVISTA PARA AVALIAÇÃO DA ABORDAGEM RSD	79
ANEXO C – TERMO DE CONSENTIMENTO DA PESQUISA	85

1. INTRODUÇÃO

A especificação de requisitos desempenha um papel fundamental no desenvolvimento de software, fornecendo uma descrição clara e detalhada dos objetivos, funcionalidades e restrições do sistema a ser desenvolvido (SOMMERVILLE, 2011).

De acordo com Thayer e Dorfman (1997), a especificação de requisitos fornece um mecanismo adequado para entender os desejos do cliente, analisando as necessidades, verificando a viabilidade, negociando soluções, especificando e gerenciando suas mudanças. Com isso, a qualidade da Especificação de Requisitos de Software é reconhecida como uma condição importante para alcançar uma vantagem competitiva na indústria de software, segundo Saito et al. (2013).

No contexto dos projetos ágeis, a especificação de requisitos se propõe a lidar com a volatilidade e a incerteza inerentes aos requisitos em constante mudança (RIBEIRO et al., 2016). O desenvolvimento ágil enfatiza a colaboração contínua entre a equipe de desenvolvimento e os clientes, priorizando a entrega de valor de forma incremental e iterativa (BECK et al., 2001). A especificação de requisitos em projetos ágeis pode ser conduzida por meio de práticas como *User Stories*, que descrevem as necessidades do usuário de forma concisa e orientada ao valor (COHN, 2004). Outras técnicas têm sido propostas para especificar requisitos em projetos ágeis, como por exemplo, InterMod (LOSADA et al., 2013), ELECTRA (RIVERO et al., 2014) e SnapMind (Wanderley et al., 2014).

De acordo com uma pesquisa conduzida pela VersionOne (2023), envolvendo cerca de 788 pessoas, 71% dos entrevistados utilizam métodos ágeis na grande parte dos projetos. No entanto, a condução das atividades de requisitos em projetos ágeis tem enfrentado desafios específicos. Dentre eles, destacam-se a necessidade de manter uma comunicação clara e efetiva entre a equipe de desenvolvimento e os *stakeholders*, garantindo um alinhamento contínuo dos requisitos (CHEN et al., 2012). Além disso, a gestão do escopo e o gerenciamento das expectativas dos clientes também são desafios relevantes nesse contexto (BROWN et al., 2011).

Na especificação de requisitos em projetos ágeis, todas as partes interessadas devem ter uma compreensão clara e compartilhada dos requisitos (HUSSAIN et al., 2015). Além disso, a especificação de requisitos deve ser suficientemente detalhada para guiar o desenvolvimento (ALVES et al., 2019).

Apesar dos obstáculos enfrentados, a especificação de requisitos em projetos ágeis tem sido amplamente adotada por organizações de diferentes tamanhos, tendo maior destaque nas organizações pequenas, as quais continuam a destacar o valor do Agile como uma estrutura organizacional e de produtividade eficaz (VERSIONONE, 2023). Essas organizações relatam uma

série de benefícios tangíveis, tais como uma colaboração mais efetiva, uma qualidade de software aprimorada e uma melhor integração com as metas do negócio.

1.1. Justificativa e Definição do Problema

Medeiros (2017) aponta que alguns problemas relacionados à atividade de especificação de requisitos podem ser tornar catalisadores para outros problemas como baixa produtividade das equipes e a dificuldade na realização de manutenção do software.

Em geral, observa-se que as especificações inadequadas e o uso de técnicas como *User Stories* contemplam poucos cenários de requisitos técnicos, sendo insuficientes para serem usadas por equipes de desenvolvimento ágeis (MEDEIROS, 2017).

Nesse contexto, a literatura apresenta algumas estratégias para atuar na resolução ou diminuição desses problemas, como por exemplo o *NLtoSTD*, que emerge como uma abordagem formal que utiliza modelos para representar requisitos de forma precisa e não ambígua. Essa técnica promove uma melhor compreensão dos requisitos ao permitir a análise e simulação dos modelos, identificando assim inconsistências, omissões e possíveis erros antes mesmo do início da implementação (ACEITUNA, 2011).

Outra abordagem é a *BehaviorMap*, que propõe especificar cenários comportamentais do usuário de forma cognitiva, com base na modelagem de mapas mentais. O objetivo é tornar os modelos de requisitos compreensíveis para os stakeholders, possibilitando uma participação mais produtiva durante o desenvolvimento de software (WANDERLEY et al., 2015).

A abordagem RSD (*Requirements Specification for Developers*) proposta por Medeiros (2017) se propõe a fornecer uma visão integrada dos requisitos funcionais e não-funcionais, proporcionando uma cobertura de requisitos mais ampla para a equipe de desenvolvimento quando comparado com outros métodos.

No entanto, a literatura apresenta uma única pesquisa de (MEDEIROS et al., 2020), que avaliou a abordagem RSD através da aplicação de estudos de casos em duas empresas, tendo os mesmos sido conduzidos pelos próprios autores da abordagem. Até o momento, não encontramos na literatura, outros estudos que tenham avaliado a abordagem RSD, apresentando seus pontos positivos e negativos. Baseado nisto, o presente trabalho busca avaliar a abordagem RSD na atividade de especificação de requisitos em um projeto ágil de uma empresa de software.

1.2. Objetivos

Objetivo geral

O objetivo principal desta pesquisa é avaliar de forma sistemática a adoção da abordagem RSD como estratégia para especificar requisitos em projetos ágeis.

Objetivos específicos

Considerando o objetivo geral proposto, delinearão-se os seguintes objetivos específicos explorados nesta pesquisa:

- Analisar como as atividades de engenharia de requisitos estão sendo realizadas atualmente em um projeto ágil em andamento;
- Implantar abordagem RSD no projeto;
- Coletar, analisar e sintetizar os dados para avaliar a adoção da abordagem RSD, considerando o impacto nas atividades de desenvolvimento da empresa, como também, uma reflexão comparativa em relação a outras abordagens utilizadas pelo projeto.

Considerando o contexto e os objetivos definidos, a seguinte questão de pesquisa foi estabelecida:

- Como a abordagem RSD é avaliada pelos *stakeholders* durante o desenvolvimento ágil de um software?

1.3. Metodologia

Esta pesquisa é de natureza aplicada, utilizando uma abordagem qualitativa, onde o pesquisador procura entender os fenômenos explorando-os, interpretando-os e relatando-os (MELLO et al., 2012). A pesquisa de natureza aplicada busca gerar conhecimento com a finalidade de resolver problemas específicos ou melhorar a prática e a tomada de decisões em um contexto real" (MACHADO et al., 2018). Por outro lado, na abordagem qualitativa, o pesquisador busca um maior entendimento dos fenômenos em análise, interpretando-os de acordo com a perspectiva dos participantes, sem atentar-se com representatividade numérica ou generalizações. Sendo assim, a interpretação do pesquisador é o principal instrumento de investigação e a necessidade de imersão no campo, para captar os significados dos comportamentos, destacando-os como características da pesquisa qualitativa (ALVES, 1991; GOLDENBERG, 1999; NEVES, 1996; PATTON, 2002; TERENCE e ESCRIVÃO, 2006).

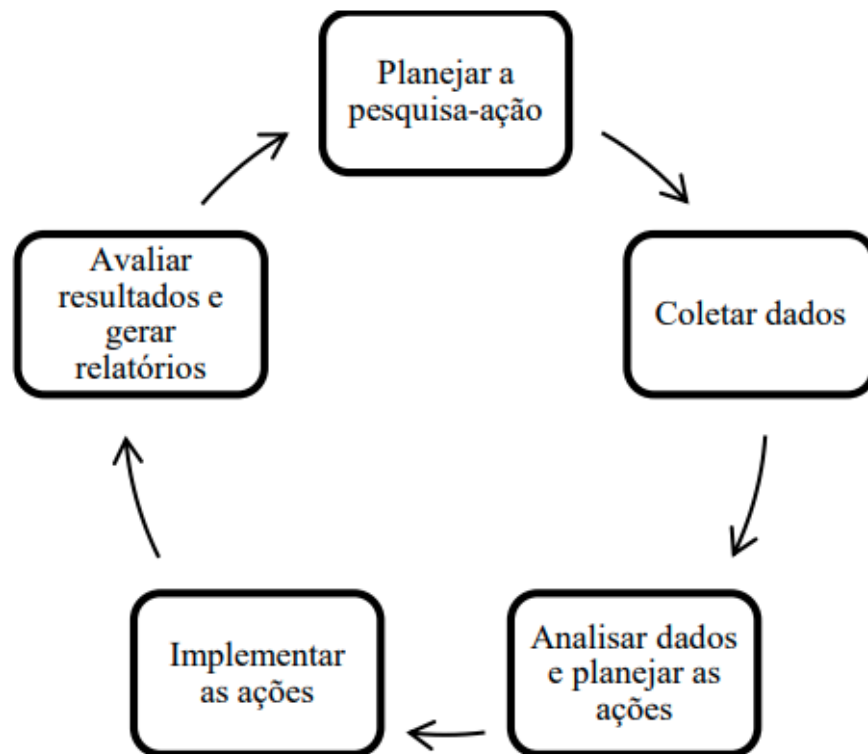
Tendo em vista a necessidade do pesquisador interferir no processo de desenvolvimento da empresa para introduzir a adoção da abordagem RSD, foi escolhido a pesquisa-ação como método de pesquisa. Segundo Mello et al. (2012), a pesquisa-ação é um método de pesquisa utilizado na engenharia de produção que busca gerar conhecimento e resolver problemas práticos. Envolve a ação nos domínios prático e de pesquisa, combinando aspectos da prática rotineira com a pesquisa científica (TRIPP, 2005).

De acordo com Westbrook (1995), Coughlan e Coughlan (2002), Ballantyne (2004), Thiollent (2007) e Mello et al. (2012), as principais características da pesquisa-ação são:

- Aplicação de abordagem científica para investigar e resolver problemas sociais ou organizacionais envolve trabalhar diretamente com aqueles que são afetados pelos problemas;
- A participação ativa e cooperativa dos membros dentro do sistema é um aspecto crucial da abordagem que está sendo usada;
- A estratégia de pesquisa visa alcançar mais do que apenas resolver problemas imediatos. Seus resultados desejados incluem valiosas lições aprendidas com o processo de pesquisa, intencional ou não, além de contribuir para o desenvolvimento da teoria e do conhecimento científico;
- Trata-se de implementar uma ação com as pessoas ou grupos envolvidos no problema em observação, com a participação ativa dos pesquisadores. Os pesquisadores desempenham um papel vital no monitoramento e avaliação das ações desencadeadas pelos problemas encontrados durante o processo. O foco é resolver os problemas em tempo real e aprender com a experiência para melhorar as respostas futuras.

Para realização desta pesquisa, foi utilizado uma adaptação do protocolo definido por Mello et al. (2012), que pode ser observado na Figura 1, a qual destaca a estrutura para condução da pesquisa-ação.

Figura 1 – Estrutura para a condução da pesquisa-ação



Fonte: Adaptado de Mello et al. (2012).

- **Planejar Pesquisa-Ação:** Esta fase teve por objetivo uma exploração inicial na empresa para descobrir o campo de estudo, identificar os interessados e suas expectativas, para isso foi realizada uma avaliação de diagnóstico da situação atual da empresa, procurando identificar os problemas prioritários e possíveis ações a serem tomadas.
- **Coletar Dados:** Esta fase teve por objetivo coletar dados sobre a utilização da abordagem RSD em um projeto na empresa. Para isso, inicialmente foi necessário realizar uma capacitação sobre a abordagem.
- **Analisar Dados:** Esta fase teve por objetivo a análise e síntese dos dados coletados durante a pesquisa utilizando o método de síntese temática, além da análise detalhada da percepção da equipe sobre a adoção da abordagem RSD e análise comparativa em relação à situação anterior do projeto.
- **Avaliar Resultados:** Esta fase teve por objetivo avaliar os resultados obtidos na pesquisa-ação, aplicando a abordagem qualitativa na realização da análise e síntese dos dados, visando responder às questões de pesquisa.

1.4. Aplicabilidade

A aplicabilidade da pesquisa vai além da empresa em estudo, oferecendo um modelo que pode ser adaptado e implementado em diversos contextos em que a agilidade e a resposta rápida às mudanças são componentes críticos para o sucesso.

A importância da aplicação desta pesquisa no contexto específico da empresa em questão reside na oportunidade de testar e validar a abordagem RSD em um ambiente real, enfrentando desafios práticos e avaliando resultados tangíveis. Isso não apenas fortalece a relevância da pesquisa para a organização envolvida, proporcionando uma base sólida para decisões estratégicas e ajustes metodológicos, mas também amplia seu valor para a comunidade acadêmica e para outras empresas que operam sob a filosofia ágil. A aplicabilidade desta pesquisa destaca-se como um esforço para conectar teoria e prática, contribuindo para o corpo de conhecimento em desenvolvimento ágil e oferecendo um guia prático para a implantação da abordagem RSD em outras empresas, incentivando uma reflexão contínua sobre os processos de desenvolvimento de software e promovendo uma cultura de inovação e melhoria contínua.

1.5. Estrutura do Documento

Os capítulos subsequentes estarão organizados da seguinte maneira: O Capítulo 2 detalha a fundamentação teórica, oferecendo um aprofundamento na especificação de requisitos em desenvolvimento ágil, além de discutir metodologias ágeis relevantes e a abordagem RSD em detalhes. No Capítulo 3, avaliamos a situação atual da empresa em foco. O Capítulo 4 aborda a implantação da abordagem, incluindo a capacitação dos envolvidos e o monitoramento da sua utilização. O Capítulo 5 dedica-se à avaliação da abordagem RSD, apresentando os resultados

obtidos e as considerações finais, enquanto o Capítulo 6 discute os resultados no contexto das questões de pesquisa, comparando-os com trabalhos relacionados e destacando as ameaças à validade da pesquisa. Por fim, o Capítulo 7 ressalta as contribuições da pesquisa e sugerindo direções para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Nas seções seguintes são apresentados conceitos da especificação de requisitos no desenvolvimento ágil de software, além de uma visão geral da abordagem RSD e os trabalhos relacionados à pesquisa.

2.1. Especificação de Requisitos no Desenvolvimento Ágil de Software

A especificação de requisitos no desenvolvimento ágil de software é uma fase crucial que serve como base para o projeto ao descrever de forma clara e precisa os objetivos, funcionalidades e limitações do sistema a ser desenvolvido, enfrentando desafios únicos devido à natureza volátil e incerta dos requisitos que estão sujeitos a mudanças frequentes (ALVES et al., 2019; RIBEIRO et al., 2016).

A eficácia da comunicação entre a equipe de desenvolvimento e os stakeholders é fundamental, sendo a manutenção de um alinhamento contínuo dos requisitos através do projeto essencial para o sucesso do desenvolvimento. A aplicação de metodologias ágeis, como *Scrum* e *Kanban*, oferece estruturas que facilitam essa comunicação por meio de reuniões regulares, revisões de sprint e o uso de quadros de visualização do projeto (HUSSAIN et al., 2015).

Gerenciar o escopo do projeto e as expectativas dos clientes é crítico, necessitando um equilíbrio entre fornecer detalhes suficientes para guiar o desenvolvimento e manter a flexibilidade para adaptações. Métodos como prototipagem rápida e revisões frequentes com *stakeholders* são vitais para a validação contínua dos requisitos (PAI et al., 2013).

A adoção da especificação de requisitos em ambientes ágeis oferece vantagens significativas, incluindo a interação constante e direta entre desenvolvedores e clientes, proporcionando um produto final mais alinhado às necessidades reais do mercado. A adaptabilidade e a resposta rápida às mudanças são particularmente valiosas em mercados voláteis e competitivos (VERSIONONE, 2021).

Técnicas avançadas de elicitação de requisitos, como *workshops* de inovação e sessões de *brainstorming*, também contribuem significativamente para o entendimento e refinamento dos requisitos em projetos ágeis, incentivando a criatividade, a colaboração e o pensamento crítico (PAI et al., 2013).

Assim, a especificação de requisitos em projetos ágeis representa uma síntese entre a necessidade de precisão e a flexibilidade para adaptar-se a mudanças, conduzindo a resultados de desenvolvimento que são inovadores e alinhados com as expectativas do cliente e as demandas do mercado.

2.2. Metodologias e Métodos Ágeis

As metodologias ágeis emergiram como uma resposta adaptativa às limitações encontradas nos métodos tradicionais de desenvolvimento de software, particularmente em projetos onde os requisitos são incertos ou sujeitos a mudanças frequentes. Conforme destacado por Sommerville (2019), uma distinção crítica entre as abordagens tradicionais e ágeis reside na flexibilidade e na capacidade de atender a uma ampla gama de demandas de desenvolvimento. Enquanto os métodos tradicionais são reconhecidos pela sua eficácia na elaboração de sistemas críticos, devido à sua ênfase na formalização rigorosa da especificação do projeto, eles frequentemente falham ao tentar desenvolver sistemas de menor escala, devido ao tempo excessivo dedicado à documentação e especificação. Este processo prolongado inevitavelmente compromete o tempo disponível para o desenvolvimento e testes, reduzindo a capacidade de resposta das equipes às mudanças (SOMMERVILLE, 2019; RAMOS, 2015).

Em oposição a este quadro burocrático, as metodologias ágeis propõem uma abordagem mais dinâmica e focada na construção efetiva do produto, minimizando a ênfase na documentação extensiva. O princípio central dos métodos ágeis é a entrega contínua de valor para o cliente, permitindo uma resposta rápida e eficaz às alterações nos requisitos e na adição de novas funcionalidades, através de ciclos de desenvolvimento curtos e iterativos (RAMOS, 2015; SOMMERVILLE, 2019).

Além disso, a adoção das metodologias ágeis facilita uma colaboração mais estreita entre as equipes de desenvolvimento e os *stakeholders*, através de práticas como desenvolvimento orientado a testes, integração contínua, programação em pares e sprints de desenvolvimento. Estas práticas não só aumentam a qualidade do produto final mas também promovem um ambiente de trabalho mais flexível e adaptável (BECK et al., 2001).

A agilidade também promove uma mentalidade de melhoria contínua, não apenas do produto em desenvolvimento mas também dos processos utilizados pela equipe de desenvolvimento. Retrospectivas regulares são uma parte fundamental das metodologias ágeis, oferecendo às equipes a oportunidade de refletir sobre o que funcionou bem e o que pode ser melhorado no próximo ciclo (SUTHERLAND et al., 2017).

Em resumo, as metodologias ágeis são projetadas para responder eficientemente à evolução das necessidades do projeto, proporcionando uma estrutura que suporta o desenvolvimento de sistemas de negócios onde os requisitos são fluidos e propensos a mudanças frequentes. Por meio da adoção de uma abordagem iterativa para as fases de especificação, desenvolvimento e entrega, os métodos ágeis permitem que as equipes produzam *software* de alta qualidade de maneira eficiente e flexível, garantindo que o produto final esteja alinhado com as necessidades e expectativas do cliente (SOMMERVILLE, 2019; RAMOS, 2015; BECK et al., 2001; SUTHERLAND et al., 2017).

Entre as metodologias ágeis mais conhecidas e amplamente utilizadas no mercado atualmente, destacam-se o Scrum (SCHWABER, 2007), *Extreme Programming* (BECK, 1999) e *Feature Driven Development* (DeLUCA et al., 2015). As subseções a seguir apresentam uma visão geral dos métodos citados, destacando suas características principais e aplicações.

2.2.1. Scrum

Segundo Schwaber (2007), o Scrum é uma estrutura que permite lidar com problemas complexos de forma adaptativa, enquanto produz produtos de maior valor. Ele não é um processo prescritivo, mas um *framework* que pode ser utilizado com diferentes processos e técnicas. Isso destaca a flexibilidade e adaptabilidade do Scrum para enfrentar a complexidade inerente ao desenvolvimento de software.

O Scrum é baseado em um ciclo iterativo e incremental chamado de *sprint*. Schwaber e Sutherland (2017) explicam que o Scrum se baseia em sprints, que permitem às equipes melhorar continuamente a aplicação e o ambiente de desenvolvimento. Cada sprint tem uma duração fixa, geralmente de duas a quatro semanas, e ao final de cada um, um incremento do produto é entregue.

Durante o *sprint*, a equipe se auto-organiza para alcançar as metas definidas. As atividades diárias são coordenadas por meio de reuniões curtas e diárias chamadas de "*daily*" ou "*stand-up*". Schwaber e Sutherland (2017) afirmam que a *daily* proporciona uma oportunidade para a equipe sincronizar suas atividades e criar um plano para as próximas 24 horas.

A transparência é uma característica fundamental do Scrum. Todos os aspectos relevantes do projeto, como o progresso, os impedimentos e as mudanças, devem ser visíveis e compartilhados com a equipe. Schwaber (2007) destaca que a transparência requer que esses aspectos sejam definidos por uma linguagem comum. Isso promove uma comunicação efetiva e permite que todos os membros da equipe tomem decisões mais informadas.

Além disso, o Scrum valoriza a colaboração e o envolvimento contínuo dos *stakeholders*. Schwaber e Sutherland (2017) afirmam que a colaboração estreita entre a equipe de desenvolvimento e o cliente é essencial para o sucesso do Scrum. Os *stakeholders* são encorajados a participar ativamente das reuniões, fornecer *feedback* e contribuir para o processo de desenvolvimento, assegurando que o produto final atenda às suas expectativas e necessidades.

2.2.2. *Extreme Programming*

Segundo Beck (1999), *Extreme Programming* (XP) é uma disciplina que se baseia em valores como simplicidade, comunicação, feedback e coragem. Esses valores são incorporados nas práticas do XP, visando promover a eficácia e a adaptação contínua.

Uma das características marcantes do XP é a programação em par, que envolve dois programadores trabalhando juntos no mesmo código. Beck (1999) destaca que essa prática resulta em uma qualidade muito superior, pois permite revisões constantes e aprimoramento mútuo. Além de melhorar o código, a programação em par promove a troca de conhecimento e a colaboração efetiva entre os membros da equipe.

Outra prática central do XP é o *Test Driven Development* (TDD), onde os testes são criados antes do código. Como Beck (2003) explica, no TDD, escreve-se um teste que descreve o comportamento esperado e, em seguida, desenvolve-se o código para passar no teste. Essa abordagem coloca os testes como guias essenciais para o desenvolvimento, garantindo que o software seja amplamente testado desde o início.

O XP também enfatiza iterações curtas e entregas frequentes de software funcional. Beck (1999) enfatiza a importância de fornecer retorno real ao cliente o mais rápido possível. Para isso, a equipe trabalha em pequenos incrementos do software, entregando versões funcionais em intervalos regulares. Essa abordagem permite que o cliente forneça feedback contínuo e que a equipe se adapte às necessidades e expectativas do cliente ao longo do projeto.

A comunicação efetiva é valorizada no XP. Beck (1999) ressalta que as pessoas são priorizadas em relação a processos ou ferramentas. Isso significa que a equipe deve manter uma comunicação aberta e constante, compartilhando conhecimentos, ideias e preocupações. Essa interação constante garante uma colaboração eficiente e um progresso contínuo no projeto.

2.2.3. *Feature Driven Development*

O *Feature Driven Development* (FDD) é uma abordagem ágil de desenvolvimento de software que se baseia na construção gradual e iterativa do software, com o objetivo de entregar recursos específicos. Conforme DeLuca e Coad (1999), o FDD é uma metodologia focada em recursos, que visa fornecer resultados visíveis e tangíveis para o cliente. Isso significa que a equipe de desenvolvimento se concentra em identificar, projetar e implementar as principais características do sistema de forma acelerada.

Uma das práticas fundamentais do FDD é a modelagem de domínio. DeLuca e Coad (1999) explicaram que a modelagem de domínio desempenha um papel crítico no FDD, pois auxilia na compreensão do problema. Nessa abordagem, a equipe trabalha em colaboração estreita

com os *stakeholders* para criar modelos de domínio que representem as características e requisitos do sistema. Esses modelos servem como base para a construção do software.

Outra prática importante do FDD é o planejamento por características. DeLuca e Coad (1999) ressaltaram que o planejamento por características é o cerne do FDD. Nesse processo, as características do sistema são identificadas e priorizadas com base no valor para o cliente. A equipe, então, define os detalhes de cada característica, estima o tempo necessário para sua implementação e atribui as tarefas aos membros específicos da equipe.

O FDD também enfatiza a inspeção frequente e a adaptação contínua. DeLuca e Coad (1999) afirmaram que o FDD é um processo iterativo e incremental, baseado na inspeção e na adaptação constantes. Isso implica que a equipe realiza verificações regulares para avaliar o progresso do projeto e ajustar o planejamento, se necessário. Essa abordagem permite que a equipe seja ágil e responsiva a mudanças durante o desenvolvimento do software.

A comunicação efetiva é valorizada no FDD. DeLuca e Coad (1999) mencionaram que a comunicação é uma parte vital do FDD, com a equipe de desenvolvimento mantendo uma comunicação constante entre si e com os *stakeholders*. A colaboração ativa e a troca de informações promovem uma compreensão clara dos requisitos e garantem uma entrega alinhada com as expectativas do cliente.

2.3. A Abordagem *Requirements Specification for Developers (RSD)*

Esta seção apresenta uma visão geral da abordagem, suas práticas de design e estrutura da abordagem, algumas possibilidades de expansão da abordagem.

De acordo com Medeiros (2017), a abordagem RSD (Requirements Specification for Developers) oferece uma visão dos requisitos, conectando de forma sistemática os conceitos do domínio do problema, por meio da modelagem conceitual; a representação visual dos requisitos de interface por meio de mockups; as regras de negócio, requisitos não-funcionais e restrições técnicas por meio de critérios de aceitação. Essas representações contemplam uma ampla gama de requisitos e estão mais próximas do campo de implementação, o que proporciona uma cobertura mais abrangente para a equipe de desenvolvimento em comparação com User Stories, por exemplo, que focam apenas em requisitos funcionais. Como resultado, a RSD permite que todos esses recursos sejam apresentados de forma única e integrada, o que facilita o entendimento dos desenvolvedores.

Com base nos valores e princípios do Manifesto Ágil (2021) que estabelece “Software em funcionamento mais que documentação abrangente”, Medeiros (2017) considera que a Especificações de Requisitos de Software (ERS) deve ser voltada para a equipe de desenvolvimento, não devendo ser utilizada como ferramenta de validação de requisitos junto ao cliente, tendo em vista que o mesmo deve validar o software funcionando. Além disso, a abordagem possui a flexibilidade de ser aplicada com *Extreme Programming (XP)*, Scrum, ou

qualquer outro método ágil, onde o cliente possa validar os requisitos através de um software funcional.

De acordo com Medeiros (2017), a abordagem propõe a utilização de três práticas de *design* bem definidas que tornam as ERSs direcionadas aos desenvolvedores: Modelagem Conceitual, Protótipo da interface do usuário e Especificação de Critérios de Aceitação+ (AC+).

Segundo Medeiros (2017), o histórico de pesquisas na literatura e na indústria de software indicam que solicitações contínuas de mudanças, são fatores que comprometem diretamente a produtividade das equipes de desenvolvimento, tendo em vista que arquiteturas inadequadas de software comumente podem apresentar problemas de escalabilidade oriundas da negligência nas atividades de projeto. Com base na abordagem RSD, é possível mitigar esses desafios por meio da modelagem conceitual, que permite identificar os principais conceitos de negócio, como entidades, atributos e relacionamentos. Essa prática auxilia na construção de uma arquitetura mais escalável e pode minimizar a necessidade de retrabalho ao lidar com mudanças frequentes. Para exemplificar o uso da RSD, a Figura 2 destaca a estrutura do artefato RSD relacionado a uma tela de cadastro de atletas.

Figura 2 – Estrutura do artefato RSD

Rótulo: Cadastro de Atleta **Prioridade:** Crítica **Recurso:** Ana **Sprint:** 1

Descrição: O sistema deverá permitir a inclusão e atualização dos dados de atletas nacionais e estrangeiros das federações esportivas reconhecidas pelo comitê Olímpico Internacional.

X

53 x 46	Nome Completo <input style="width: 90%;" type="text"/>	Último Nome <input style="width: 80%;" type="text"/>	Iniciais <input style="width: 70%;" type="text"/>
Foto <input style="width: 60px; height: 20px;" type="text"/>	Data Nascimento <input style="width: 60px;" type="text"/>	Idade <input style="width: 30px;" type="text"/>	Gênero <input style="width: 60px;" type="text"/>
	<input type="checkbox"/> Estrangeiro	Número Passaporte <input style="width: 100px;" type="text"/>	
E-mail <input style="width: 100%;" type="text"/>			Telefone <input style="width: 80px;" type="text"/>
Confederação <input style="width: 100%;" type="text"/>			Número de Registro <input style="width: 100px;" type="text"/>
<input type="button" value="Salvar"/>			

^
Protótipo

Widget	Conceitos	Critérios de Aceitação
Foto	person.name	AC09
Nome Completo	person.fullName	-
Último Nome	person.lastName	-
Iniciais	athlete.initialsName	AC13, AC90
Data Nascimento	athlete.birthDate	AC05, AC06
Idade	-	AC07, AC90
Gênero	person.gender	-
Estrangeiro	person.isForeign	AC17
Número Passaporte	person.passportNumber	AC17
E-mail	person.email	AC01
Telefone	person.phone	-
Confederação	confederation.name	AC04, AC20
Número de Registro	athlete.registerNumber	AC21
Salvar	-	AC03, AC08
-	person.idPerson	AC12

^
Componentes do Protótipo

^
Entidades de Dados

^
AC+

Fonte: Adaptado de Medeiros (2017).

Entende-se que além da descrição da interface, é feito o uso de protótipos de média fidelidade, para definir como a página deve ser implementada, além de descrever os critérios de aceitação ligados à aplicação, e mais especificamente a cada *widget*.

Como podemos observar na Figura 2, um artefato RSD é definido em cinco partes (MEDEIROS, 2017):

- Descrição de alto nível: Esta seção deverá incluir rótulo, descrição de alto nível, prioridade, parte interessada solicitante e sprint em que será implementado.
- Protótipo: Se aplicável, o protótipo de média fidelidade permite a visualização dos dados e como eles serão apresentados no sistema.
- *Widget*: Apresenta o widget (componente) presente no protótipo.
- Entidades de dados: Apresenta as entidades de dados e atributos extraídos do modelo conceitual, que estão relacionados a cada widget.
- Critérios de aceitação: Apresenta a identificação (ID) dos Critérios de Aceitação. Os Critérios de Aceitação devem ser reutilizados na Abordagem RSD. Eles são escritos em outra tabela e referenciados por ids nos artefatos.

Nas próximas subseções apresentamos um detalhamento sobre as três práticas da abordagem RSD.

2.3.1 Modelagem Conceitual

Tradicionalmente, quando se define um sistema de informação, consideram-se duas perspectivas: uma relacionada à sua função e outra relacionada à sua estrutura (HIRSCHHEIM, 1995). A perspectiva funcional busca descrever a funcionalidade do sistema de forma geral, levando em conta o ponto de vista do usuário. Já a perspectiva estrutural representa o sistema em termos de entidades, relacionamentos estáticos e restrições. A modelagem conceitual é uma atividade sistemática que visa descrever certos aspectos do mundo físico e social, com o objetivo de facilitar a compreensão e a comunicação (LOUCOPOULOS, 1992). De acordo com Olivé (2007), modelos conceituais são necessários para que todas as partes interessadas possam ter uma compreensão comum do domínio do sistema.

No entanto, embora a modelagem conceitual seja uma prática comum em abordagens tradicionais, ela não é usada de maneira sistemática em métodos ágeis populares, como Scrum (SCHWABER, 2001) e XP (BECK, 1999). As abordagens ágeis de requisitos, como *User Stories*, focam principalmente na modelagem funcional. Pouca importância é dada à definição sistemática do modelo conceitual junto com os *User Stories*. Em alguns casos, as entidades de dados são derivadas das classes definidas no código-fonte, o que pode levar a um modelo de dados instável. Um modelo conceitual inadequado pode dificultar a inclusão de novos recursos, a incorporação de

mudanças e o fornecimento de dados para sistemas de inteligência empresarial. Algumas alterações no modelo conceitual podem ter um impacto significativo no código. Portanto, é importante dedicar tempo ao projeto do modelo conceitual. Embora Cohn (2004) proponha a identificação de conceitos (atores) e informações sobre o perfil de cada ator, essa abordagem não aborda outros aspectos importantes no contexto empresarial nem a relação entre esses atores, tratando apenas dos atores em si.

Segundo Medeiros (2017), a modelagem conceitual não é uma prática comum nos métodos ágeis, exceto no *Feature Driven Development* (FDD), que é um método ágil em que os requisitos funcionais são descritos por meio de características e diagramas de sequência, e a modelagem conceitual é utilizada por meio de um modelo de objetos que evolui para um modelo de classes com atributos e operações. No entanto, o FDD aborda apenas requisitos funcionais, deixando de lado os requisitos não-funcionais e outros requisitos do sistema.

Medeiros (2017) destaca que a abordagem RSD busca integrar as perspectivas funcional e estrutural para obter uma visão única do sistema. Os critérios de aceitação (AC+) definem requisitos funcionais, aspectos não funcionais e aspectos técnicos sob a perspectiva do desenvolvedor. A perspectiva estrutural é obtida por meio da representação dos conceitos (entidades) e seus relacionamentos, sem a definição de operações. Quando aplicável, o protótipo da interface do usuário interconecta as outras duas perspectivas funcional e estrutural.

2.3.2 Prototipação da Interface do Usuário

Protótipos de interface do usuário, também conhecidos como *mockups* ou *wireframes*, são representações visuais que demonstram como a interface de um software deve ser utilizada durante a interação com o usuário final. Esses *mockups* podem variar desde desenhos simples, utilizados para apresentar as interações entre o usuário e o sistema, até versões mais detalhadas com elementos gráficos mais elaborados, como logotipos específicos ou cores relacionadas à marca (RICCA, 2014). Medeiros (2017) destaca que embora existam diversas ferramentas disponíveis para criar *mockups* digitais, muitos profissionais ainda preferem esboçar esses protótipos em papel.

Os *mockups* têm se mostrado uma prática eficaz para capturar e definir os requisitos funcionais de um sistema (INAYAT, 2015; RICCA, 2014). Eles contribuem para a coleta de requisitos de forma eficiente, sem demandar um esforço adicional no processo. Uma das principais vantagens dos *mockups* é que eles são valiosos do ponto de vista técnico para os desenvolvedores, ao mesmo tempo em que são facilmente compreensíveis pelos usuários finais (RIVERO, 2014).

Embora muitos projetos ágeis exijam o *design* da interação do usuário, a integração de *mockups* no Desenvolvimento Ágil de Software (DAS) ainda não é amplamente compreendida

(NEBE, 2016). Segundo Medeiros (2017), os métodos ágeis populares geralmente não incluem os *mockups* como parte de seu processo, com exceção do *Dynamic Systems Method* (DSDM), que também é um dos métodos ágeis cujos autores contribuíram para a definição do Manifesto Ágil (2001). No entanto, o DSDM é considerado o método ágil mais complexo, envolvendo treze papéis e a produção de quatorze artefatos ao longo do processo de desenvolvimento. Como resultado, sua adoção na indústria tem sido relativamente baixa em comparação com outros métodos, como Scrum e XP (VERSIONE, 2016). Portanto, as empresas precisam adaptar seus processos para integrar os *mockups* no DAS (MEDEIROS, 2017).

Medeiros (2017) destaca que a técnica proposta utiliza os *mockups* como uma forma de interconectar os critérios de aceitação (AC+) e os modelos conceituais, facilitando assim a compreensão por parte dos desenvolvedores.

2.3.3 Critérios de Aceitação

Um das características fundamentais da abordagem RSD são os Critérios de Aceitação + (AC+), que são uma ampliação do conceito de Critérios de Aceitação. O AC+ estabelece alguns critérios além das regras de negócio, como regras de validação, interface, técnicas e qualquer outro tipo de restrição indispensáveis para a construção do software, e podem ser classificados em seis categorias (MEDEIROS, 2017):

- Negócio (N): Uma ou mais regras relacionadas à natureza do negócio.
- Validação (V): Alguma validação que é requerida do software, mas não está estritamente relacionada ao negócio.
- Interface (I): Qualquer regra relacionada ao usuário da interface.
- Técnico (T): Uma ou mais restrição técnica sobre como a solução deve ser implementada na aplicação.
- Não Funcional (NF): Preocupações sobre rastreabilidade.
- Outro (O): Quando não se enquadra em uma das categorias descritas.

Um requisito de software pode conter inúmeros AC+, contendo diferentes prioridades que também podem ser destinadas para diferentes *sprints*. Novos AC+ também podem ser identificados ao longo do processo de desenvolvimento. Alguns exemplos de AC+ são apresentados na Tabela 1. Esses critérios estão relacionados ao “Cadastro de atleta”, citado na Figura 2, no início desta seção. A coluna Tipo refere-se a um dos seis tipos de categoria. A última coluna define se o AC+ é da categoria Geral (G) porque tem potencial para ser reaproveitado, ou se é Específico (S) de um requisito.

Tabela 1 – Exemplos de critérios de aceitação.

ID	Descrição	Tipo	Categoria
AC01	O endereço de e-mail deve ser válido.	V	G
AC03	Para salvar é necessário que todos os campos obrigatórios (*) sejam preenchidos.	V	G
AC04	Somente registros ativos devem ser exibidos.	V	G
AC07	A idade deve ser calculada a partir da data de nascimento.	V	G
AC08	A rotina para salvar um atleta também deve salvar os endereços correspondentes.	T	S
AC09	A operação de leitura e gravação de arquivos no sistema de arquivos deve ser feita através de endereço relativo.	T	G
AC12	O código sequencial para identificar o registro deve ser gerado pelo banco de dados.	T	G
AC13	As iniciais do atleta deverão ser extraídas do nome do atleta, por exemplo, se o nome for “Fabiana de Almeida Murer”, as iniciais deverão ser “F.A.M”.	T	S
AC17	Todos os atletas estrangeiros deverão possuir número de passaporte.	N	S
AC20	A lista suspensa deve exibir apenas as confederações que o usuário logado tem permissão de acesso em seu perfil.	N	G
AC21	Não pode haver dois atletas com o mesmo número de inscrição na mesma confederação.	N	S
AC50	A etiqueta deverá utilizar o recurso multilíngue.	NF	G
AC90	O <i>widget</i> é somente leitura. Não pode ser alterado pelo usuário.	I	G

Fonte: Adaptado de Medeiros (2017).

O AC+ pode ser reutilizado por diferentes requisitos e possuem associação com identificação única para possibilitar a reutilização. Por outro lado, a reutilização também pode ocorrer no mesmo RSD. Em certos casos, no artefato RSD, dois ou mais componentes (*widgets*), como botões, caixas de texto, inputs, entre outros, podem compartilhar o mesmo critério. Alguns AC+ não possuem relação com *widgets*, como o AC+ em relação à implementação de algoritmos ou requisitos não funcionais. Nesses casos específicos, eles devem ser adicionados em uma linha com a coluna '*widget*' em branco. Alguns *widgets* também poderão não estar relacionados a alguma entidade de dados e também deverão ser adicionados em uma linha com a coluna 'Conceitos' em branco (MEDEIROS, 2017).

Medeiros (2017) destaca que a aplicação das práticas do RSD pode ser adaptada de acordo com o tipo de projeto, software desenvolvido ou para atender a qualquer particularidade das equipes de desenvolvimento. A modelagem de protótipos se encaixa perfeitamente em alguns tipos de projetos, como desenvolvimento de sistemas de informação e aplicativos móveis. No entanto, alguns softwares não possuem uma interação visual significativa com o usuário, como serviços da web e aplicativos de processamento em lote. Nesses casos, a abordagem do RSD pode estar mais focada na modelagem dos conceitos e na especificação dos AC+. O mesmo se aplica a algumas empresas que utilizam frameworks para gerar uma interface de usuário básica a partir do modelo de dados.

Alguns projetos têm como objetivo a manutenção e evolução de aplicações existentes. Geralmente, esses projetos possuem um modelo de dados estável, portanto, as atividades de desenvolvimento se concentram na inclusão de novos recursos e correção de bugs. Nesse tipo de projeto, que ainda não possui um RSD e a aplicação já está em produção, o uso da abordagem pode ser simplificado para descrever apenas os novos conceitos, *widgets* e AC+ e aqueles que serão alterados. Em vez de descrever todos os *widgets* e conceitos que já estão implementados no software (MEDEIROS, 2017).

De acordo com Medeiros (2017), a abordagem sugere que o AC+ seja classificado em alguns tipos para incentivar a reutilização, conforme definido na seção anterior. Essa classificação pode e deve ser adaptada de acordo com as particularidades de cada empresa.

2.4. Trabalhos Relacionados

Os trabalhos relacionados foram identificados por meio de uma busca exploratória realizada no Google Acadêmico¹, com ênfase em pesquisas que avaliaram a utilização de alguma abordagem de especificação de requisitos. Três estudos principais se destacaram nessa busca: (ACEITUNA, 2011), (WANDERLEY et al., 2015) e (MEDEIROS et al., 2020).

Segundo Aceituna et al. (2011) o método NLtoSTD, tem como principal objetivo melhorar a qualidade das especificações de requisitos escritas em linguagem natural, enfrentando desafios como a ambiguidade e a incompletude inerentes a essa forma de representação. Para atacar esses desafios, a abordagem transforma requisitos em linguagem natural *Natural Language* (NL) em um diagrama de transição de estados *State Transition Diagram* (STD), permitindo a verificação por meio de raciocínio automatizado. Esse método visa expor ambiguidades e funcionalidades ausentes nos requisitos, oferecendo *insights* e *feedbacks* para aperfeiçoar tanto o método quanto a qualidade dos requisitos.

O resultado obtido, conforme reportado no estudo, mostrou que o método NLtoSTD é capaz de identificar ambiguidades e funcionalidades ausentes em um conjunto de requisitos em

¹ <https://scholar.google.com.br>

linguagem natural. O estudo, conduzido com participantes na *North Dakota State University*, evidenciou que a abordagem proposta tem potencial para expor defeitos nos requisitos que métodos tradicionais de inspeção podem não detectar facilmente, especialmente em casos de requisitos de alta coesão.

Por outro lado, em Wanderley et al. (2015) a abordagem *BehaviorMap* propõe a especificação de cenários comportamentais do usuário de maneira cognitiva, utilizando a modelagem de mapas mentais. O intuito é tornar os modelos de requisitos compreensíveis para os stakeholders, promovendo uma participação mais produtiva durante o processo de desenvolvimento de software.

Um estudo experimental foi realizado para avaliar o esforço cognitivo na compreensão de cenários *BehaviorMap* em comparação com cenários textuais tradicionais. Os resultados revelaram que os cenários da abordagem são promissores, demonstrando melhor desempenho em comparação com os cenários textuais, especialmente em cenários de alta complexidade, onde diferenças significativas a favor da abordagem foram observadas. Esses achados sugerem que a abordagem pode ser uma ferramenta valiosa para aprimorar a compreensão e a comunicação dos requisitos de software entre os stakeholders e as equipes de desenvolvimento. (WANDERLEY et al., 2015).

Além dos trabalhos relacionados identificados, a pesquisa *Requirements specification for developers in agile projects: Evaluation by two industrial case studies* (MEDEIROS et al., 2020), teve o objetivo de avaliar atualização da abordagem RSD por meio de dois estudos de caso realizados em projetos industriais ágeis.

De acordo com Medeiros et al. (2020), os resultados da avaliação apontaram que as práticas utilizadas na abordagem de especificação de requisitos para desenvolvedores têm um potencial significativo para criar uma especificação de requisitos mais objetiva e adaptada à equipe de desenvolvimento. O artigo também discutiu a opinião de profissionais que relataram que a abordagem RSD resultou em uma melhoria na integridade e legibilidade dos requisitos de software em comparação a técnicas como User Stories.

Comparando os trabalhos, percebe-se uma evolução nas abordagens de especificação de requisitos, cada uma abordando diferentes desafios inerentes à compreensão e comunicação eficaz de requisitos. A abordagem NLtoSTD foca na precisão e na eliminação de ambiguidades. *BehaviorMap*, por sua vez, aborda a compreensão cognitiva, utilizando mapas mentais para facilitar o entendimento dos stakeholders. Finalmente, a abordagem RSD adapta-se especificamente às necessidades de projetos ágeis, integrando requisitos de forma mais abrangente e orientada para o desenvolvimento.

Essa pesquisa se diferencia desses trabalhos ao conduzir uma avaliação mais ampla e detalhada de uma abordagem de especificação em um projeto ágil real de uma empresa de

software. Embora o estudo de Medeiros et al. (2020) também tenha sido conduzido em um ambiente ágil, essa pesquisa adota uma metodologia de pesquisa-ação, permitindo uma intervenção direta e um acompanhamento contínuo durante a avaliação, fornecendo evidências valiosas sobre todo o processo de implementação e utilização da abordagem RSD, além dos desafios enfrentados, contribuindo com uma análise crítica e detalhada da abordagem em um ambiente dinâmico e em rápida mudança, característico dos projetos ágeis.

2.5. Considerações Finais

Ao final deste capítulo, foi estabelecida uma base teórica que não apenas evidencia o potencial da abordagem RSD para o desenvolvimento ágil de software, mas também reforça a necessidade de pesquisas adicionais neste domínio. Importante destacar que, ao analisar os trabalhos relacionados, fica evidente a relevância dessa pesquisa no contexto atual, onde práticas eficazes de especificação de requisitos são cruciais. Este capítulo serve como alicerce para os próximos, onde metodologias e resultados práticos serão explorados, unindo teoria à aplicação direta no ambiente de desenvolvimento ágil.

3. AVALIAÇÃO NA EMPRESA ANTES DA IMPLANTAÇÃO DA ABORDAGEM RSD

O objetivo principal deste capítulo é apresentar os resultados da avaliação realizada em uma empresa antes da implantação da abordagem RSD. Entretanto, antes disso, o capítulo apresenta o protocolo de Pesquisa-Ação e o planejamento feito para a realização da pesquisa-ação, incluindo também o planejamento das etapas de implantação e avaliação da Abordagem RSD.

3.1. Protocolo de Pesquisa-ação

O protocolo de pesquisa-ação elaborado, delinea de forma clara o plano de ação que será seguido para avaliar a eficácia da abordagem RSD. O objetivo deste protocolo é orientar a pesquisa de forma a atingir os objetivos propostos, conduzindo as fases, etapas e atividades de forma sistemática.

O protocolo para esta pesquisa incorpora as seguintes etapas:

1. Introdução: Exploração do objetivo de avaliar a abordagem RSD em projetos ágeis por meio de pesquisa-ação qualitativa.
2. Planejamento da Pesquisa-Ação: Identificação do problema e objetivos, exploração e coleta de dados inicial para entender o contexto, estabelecimento da estrutura teórica-conceitual.
3. Implantação da Abordagem: Capacitação dos envolvidos na abordagem RSD, aplicação da abordagem em um projeto ágil em andamento, monitoramento do uso e artefatos criados.
4. Coleta de Dados: Entrevistas semiestruturadas respondidas por desenvolvedores e gerente de projeto, observação das atividades e artefatos de especificação de requisitos.
5. Avaliação dos Resultados: Análise e síntese dos resultados, análise da percepção da equipe e comparação com dados obtidos antes da aplicação da abordagem.
6. Conclusão e Relato dos Resultados: Sistematização dos resultados, destaque de lições aprendidas e elaboração de relatório final.

É crucial destacar que a contribuição significativa deste protocolo reside na sua capacidade de servir como referência fundamental para esta pesquisa e pesquisas futuras de natureza semelhante. Ao fornecer uma estrutura detalhada, o protocolo não apenas endereça desafios imediatos, mas também estabelece bases sólidas para enriquecer o conhecimento científico e a prática no domínio da Engenharia de Software ágil.

3.2. Iniciação do Projeto de Pesquisa-Ação

A fase de iniciação do projeto de pesquisa-ação se deu com uma avaliação da situação da empresa no que diz respeito às atividades de requisitos utilizadas para desenvolver software. Esta fase teve por objetivo estabelecer as bases conceituais e práticas necessárias para a avaliação sistemática da abordagem RSD, identificando os procedimentos metodológicos e os critérios de avaliação relevantes para a condução da pesquisa. De acordo com Mello et al. (2012), as etapas apresentadas nas subseções 3.2.1, 3.2.2 e 3.2.3 são sugeridas como parte fundamental do processo de iniciação do projeto de pesquisa-ação.

3.2.1. Definição da Estrutura Conceitual-Teórica

A estrutura conceitual desta pesquisa foi composta da questão de pesquisa e dos objetivos apresentados no Capítulo 1. A fundamentação teórica e trabalhos relacionados estão detalhados ao longo do Capítulo 2.

3.2.2. Definição da Unidade de Análise

O objeto de estudo desta pesquisa é um projeto ágil em andamento de uma empresa de software brasileira, que oferece serviços para infraestrutura, desenvolvimento de integração e microsserviços. Fundada em 2015, a empresa está localizada em Pelotas, Rio Grande do Sul e possui funcionários que trabalham em várias cidades do Brasil.

De fato, o objeto de estudo é a ERS em um projeto de e-commerce que está em fase de desenvolvimento. O escopo da aplicação vai desde o cadastramento de novas contas, gerenciamento de carteiras de clientes, cadastro de vendedores e clientes, e a venda de produtos de diferentes categorias. O projeto pertence a uma empresa brasileira, líder nos segmentos de refrigeração, climatização e eletros.

Cabe ressaltar ainda, que embora o projeto e suas atividades de ERS façam parte do objeto de estudo, esta pesquisa tem como objetivo principal avaliar a abordagem de RSD através do time de desenvolvimento do projeto. Sendo assim, ao final do trabalho os principais resultados serão fornecidos a partir da equipe de desenvolvimento.

Faz-se necessário destacar que a identidade da empresa e stakeholders será preservada.

3.2.3. Definição do Contexto e Propósito

Esta seção tem o objetivo de definir de maneira objetiva o contexto e propósito da pesquisa.

3.2.3.1. Definição do Tema e Interessados

De acordo com Mello et al. (2012), nesta etapa deve-se definir o grupo de pesquisadores e membros da unidade de análise que participarão de forma colaborativa na realização da pesquisa, coleta de dados e execução de ações para o enfrentamento do problema de pesquisa identificado.

Para conduzir a pesquisa, foram identificados os indivíduos que possuíam as competências e habilidades necessárias com base no objetivo principal do estudo. A Tabela 2 fornece informações sobre os indivíduos envolvidos na pesquisa, incluindo a quantidade e funções atribuídas.

Tabela 2 – Indivíduos envolvidos na pesquisa.

Quantidade	Função
1	Gerente de Projeto
1	Product Owner
1	Tech Lead - Pesquisador
2	Desenvolvedor de Software - Sênior
2	Desenvolvedor de Software - Pleno
3	Desenvolvedor de Software - Junior
4	Analistas (Time funcional do cliente)

3.2.3.2. Delimitação do Problema

A seção 1.1 deste trabalho estabelece o terreno para a exploração dos desafios enfrentados na especificação de requisitos dentro de projetos de desenvolvimento ágil de software. A abordagem ágil, apesar de promover uma maior adaptabilidade e resposta às mudanças, pode revelar lacunas na especificação de requisitos que podem levar a problemas como redução da produtividade das equipes e dificuldades na manutenção de software. O uso predominante de User Stories, embora benéfico para capturar requisitos funcionais de forma concisa, frequentemente não abrange totalmente os requisitos técnicos necessários para o desenvolvimento (MEDEIROS, 2017; SOMMERVILLE, 2011; RIBEIRO et al., 2016).

Nesse contexto, a abordagem RSD, emerge como uma solução potencial ao oferecer uma visão integrada dos requisitos funcionais e não-funcionais. No entanto, a literatura existente apresenta uma carência de estudos avaliativos sobre a eficácia dessa abordagem em ambientes reais. Portanto, esta pesquisa visa não apenas implementar a abordagem RSD em um contexto prático, mas também avaliar sua efetividade e impacto nas atividades de desenvolvimento e na percepção da equipe de desenvolvimento.

Conforme definido anteriormente na seção 1.2, a questão central desta pesquisa é: "Como a abordagem RSD é avaliada pelos stakeholders durante o desenvolvimento ágil de um software?". Isso nos leva a investigar, em profundidade, os benefícios e possíveis limitações da RSD, buscando contribuições significativas para as práticas de engenharia de software ágil e para o corpo de conhecimento acadêmico na área.

A implantação e avaliação da abordagem RSD são descritas em detalhes nos capítulos 4 e 5, respectivamente. Neste capítulo, abordamos a avaliação da situação da empresa, antes da implantação da abordagem RSD.

3.2.3.3. Definição dos Critérios de Avaliação da Pesquisa-Ação

Para assegurar a eficácia da pesquisa-ação em nosso estudo, estabelecemos critérios de avaliação claros e objetivos. Esses critérios são fundamentais para avaliar se a pesquisa-ação atendeu às suas metas de promover entendimento, gerar e assimilar conhecimento, e alcançar a contribuição esperada.

Primeiro Critério: Avaliar se a pesquisa-ação facilitou um entendimento comum do projeto entre todos os envolvidos, incluindo a equipe de desenvolvimento e os stakeholders. Isso foi medido através de entrevistas semiestruturadas e observações diretas, focando na clareza dos objetivos do projeto, eficácia da comunicação interna e compreensão dos requisitos da abordagem RSD.

Segundo Critério: Avaliar a capacidade da pesquisa-ação de gerar novo conhecimento e facilitar sua assimilação pelos envolvidos. Este critério foi avaliado examinando a produção de artefatos e a aplicabilidade das soluções desenvolvidas no contexto do projeto, refletindo sobre como os conhecimentos foram aplicados na prática e a capacidade da equipe de incorporar novas práticas e soluções.

Terceiro Critério: Medir a contribuição da pesquisa-ação para o campo do desenvolvimento ágil de software, especialmente em relação à especificação de requisitos. Foi avaliado o feedback dos participantes sobre a utilidade e aplicabilidade da abordagem RSD, além da aderência das práticas implementadas com as melhores práticas recomendadas.

Ao definir esses critérios de avaliação, buscamos não apenas compreender os efeitos imediatos da implementação da abordagem RSD em um projeto ágil específico, mas também avaliar seu impacto a longo prazo e sua relevância para a comunidade de desenvolvimento de software ágil como um todo. A avaliação desses critérios é apresentada na seção 5.3.

3.3. Procedimentos para Coleta, Análise e Síntese dos Dados

Esta subseção descreve os procedimentos adotados para coletar, analisar e sintetizar os dados utilizados na avaliação da situação da empresa antes da implementação da abordagem RSD, bem como na avaliação da abordagem após sua aplicação.

De acordo com Coughlan e Coughlan (2002), o pesquisador obtém dados engajando-se ativamente nos processos organizacionais do dia-a-dia relacionados ao projeto de pesquisa-ação. Os dados não são obtidos apenas por meio da participação e observação das equipes no trabalho, resolução de problemas e tomada de decisões, mas também por meio de intervenções feitas para avançar na pesquisa. Algumas dessas observações e intervenções são feitas formalmente, por meio de reuniões e entrevistas.

De acordo com a perspectiva de Mello et al. (2012), os dados podem ser divididos em dois tipos: primários e secundários. Os dados primários são coletados por meio de estatísticas operacionais, relatórios financeiros e relatórios de marketing, enquanto os dados secundários são obtidos por meio de observações, discussões e entrevistas.

Esta pesquisa utilizou principalmente dados primários, os quais foram coletados por meio de entrevistas técnicas semiestruturadas e observação direta da equipe. As fontes desses dados incluíram métodos como entrevistas técnicas semiestruturadas e observação direta da equipe.

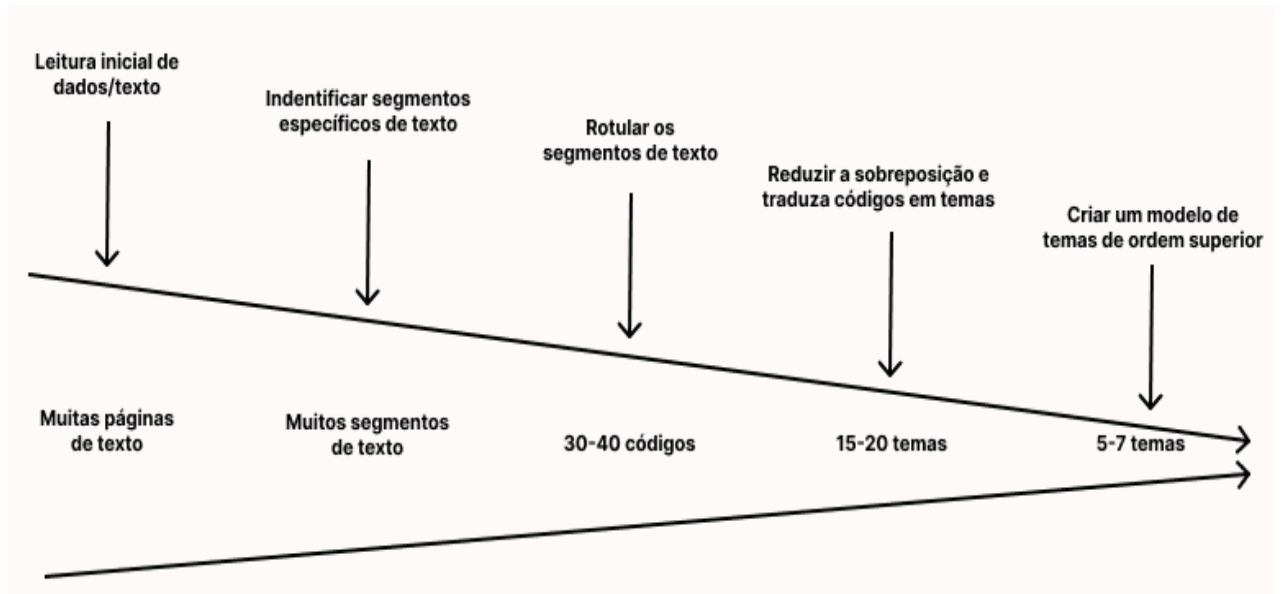
Para a coleta de dados, foram desenvolvidos dois roteiros detalhados, conforme indicado nos Apêndices A e B, com o propósito de orientar as entrevistas, para avaliar a percepção dos stakeholders antes e após a implantação da abordagem RSD no processo de desenvolvimento de um software. A estratégia utilizada seguiu um formato de funil, iniciando com perguntas de natureza geral e progredindo para inquirições mais específicas, conforme preconizado por Runeson e Martin (2009). Para assegurar a eficácia dos roteiros, foi realizado um teste piloto, o qual possibilitou a validação do roteiro da entrevista.

Com a obtenção da autorização da empresa para o contato direto com cada participante do projeto, iniciou-se o contato preliminar por meio da ferramenta *Teams*, convidando-os a participar do estudo. As entrevistas foram individualmente agendadas e antes de cada sessão, o propósito do estudo foi explicitado aos participantes, enfatizando a importância de fornecer respostas detalhadas. Na primeira fase de coleta de dados foi utilizado o roteiro do Apêndice A para realização das entrevistas, onde além do registro das respostas, as observações pertinentes dos entrevistados também foram registradas.

Para iniciar a análise e síntese dos dados, foi utilizado o método de síntese temática, que envolveu a organização e análise de dados através da identificação e agrupamento dos temas e padrões recorrentes. Seguimos o processo preconizado por Cruzes e Dyba (2011), exemplificado na Figura 3.

De maneira resumida, o procedimento se inicia com uma leitura preliminar dos dados, seguida pela identificação e rotulação de segmentos específicos por meio de códigos. Esses códigos são posteriormente traduzidos em temas, e então, identificam-se temas de nível mais abrangente (categorias) para estabelecer o modelo que aborda as questões de pesquisa.

Figura 3 – Processo de síntese temática



Fonte: Adaptado de Cruzes e Dyba (2011).

Após a realização das entrevistas, foi dado início ao processo de transcrição usando a ferramenta de inteligência artificial *Sonix*², buscando criar um registro textual abrangente das interações capturadas. As transcrições foram cuidadosamente revisadas para garantir precisão e fidelidade às respostas dos participantes. Uma leitura integral das transcrições foi conduzida para obter uma visão aprofundada dos dados coletados.

A etapa seguinte envolveu a codificação aberta, na qual foram identificados e rotulados os conceitos, ideias ou padrões relevantes presentes em cada entrevista. Os códigos resultantes foram então agrupados em temas, destacando suas semelhanças e relações intrínsecas. Nesta fase, foram identificados códigos que representam conceitos, ideias e padrões recorrentes nas respostas dos desenvolvedores. Exemplos de códigos incluem "Consolidação dos dados", "Critérios objetivos" e "Flexibilidade".

Os códigos identificados foram agrupados em temas relevantes. Por exemplo, os códigos relacionados à consolidação dos dados foram agrupados no tema "Objetividade". Isso destacou a importância da objetividade e consolidação das informações para o time de desenvolvimento.

Temas como "Especificação voltada para o cliente" e "Falta de objetividade dos critérios de aceitação" emergiram de maneira consistente nas entrevistas iniciais, indicando elementos-chave na identificação do estado inicial do projeto.

² <https://sonix.ai/pt>

Os temas identificados foram organizados em categorias mais amplas, como "Conteúdo" e "Estrutura", fornecendo evidências claras sobre as percepções da equipe em relação ao projeto e aplicação da abordagem.

3.4. Resultados da Avaliação

Nesta seção, são apresentados os resultados detalhados da avaliação realizada no contexto do projeto de pesquisa-ação, focando no detalhamento e na análise das práticas existentes e nas percepções dos participantes. Os dados foram coletados através de entrevistas semiestruturadas e observação direta, com o objetivo de identificar tanto os pontos positivos eficazes quanto os desafios enfrentados no ambiente de desenvolvimento ágil de software.

3.4.1. Processo de Especificação Utilizado no Projeto Antes da Implantação da Abordagem RSD

Antes da implantação da abordagem RSD, as histórias de usuário, critérios de aceitação e protótipos eram descritas em um item de backlog do produto, usando a ferramenta de gerenciamento de projeto *Azure DevOps*³, como mostra a Figura 4.

Figura 4 – Item de backlog do produto

PRODUCT BACKLOG ITEM 26913*

26913 [Carrinho] Proposta de Negócio com desconto em frete (arbitragem)

No one selected 9 Comments AX X Back-end X OCC X OIC X QA X Refinado X Release.15 X +

State ● Development Area NOVO-PETRA

Reason 🏠 Moved out of state Specified Iteration NOVO-PETRA\Sprint 26

Description

Eu como vendedor quero arbitrar o preço do frete do pedido para compor a melhor negociação para o cliente em atendimento.

Link XD: <https://xd.adobe.com/view/bdcb6d9b-4248-4600-8a0c-6c9ee9b191d6-e898/screen/e4bc0411-1d20-420c-8268-1627db862866> [telas 12 a 14]

Link explicativo da parte de PN de Frete para DEV e QA: <https://xd.adobe.com/view/db518077-cc0c-497c-baca-b9d74287879d-8e02/screen/f2c80f12-dad3-4ed7-ba1a-28cf70b46b71?fullscreen>

Critérios de Aceite

1. Clientes autorizados podem pedir arbitragem de frete
 1. Cliente autorizado a arbitragem de frete estará cadastrado no ERP no módulo "VE12 Parâmetro Gerais" > submódulo "Cadastro de Clientes que Arbitram Frete"
2. SE cliente em atendimento com autorização de arbitragem de frete, sistema deve
 1. No carrinho, na aba "Entrega"
 1. Apresentar flag "Negociar frete" para cada filial de origem (shipping group)
 1. Todas as flags de "Negociar frete" devem aparecer deselecionadas

³ <https://azure.microsoft.com/pt-br/products/devops>

As histórias de usuário eram escritas diretamente a partir de conversas entre a consultoria e o time funcional do cliente. Uma funcionalidade era dividida em várias histórias de usuário, e os desenvolvedores criavam tarefas para desenvolvê-las.

No decorrer de cada sprint, a equipe desenvolvia as funcionalidades e posteriormente, as versões eram liberadas em um ambiente de homologação para validar os critérios de aceitação de cada história de usuário. Uma equipe de testes realizava as validações e aprovava a conclusão e a apresentação da funcionalidade para o cliente durante a Review.

3.4.2. Pontos Positivos Identificados

Durante as entrevistas semiestruturadas, diversos pontos positivos foram destacados pelos entrevistados em relação às práticas adotadas no projeto:

- **Participação Diária do Cliente:** A equipe valorizou a participação diária do cliente como um aspecto positivo, facilitando a comunicação e o alinhamento de expectativas.
- **Adoção de Sprints Curtas:** O uso de sprints de duas semanas foi mencionado como eficaz para o desenvolvimento iterativo e entrega contínua de funcionalidades.
- **Integração de Clientes nas Squads:** A integração dos clientes nas squads foi percebida como benéfica, promovendo uma comunicação frequente e apropriada que contribuiu para a qualidade das entregas.
- **Utilização de Ferramentas como Azure DevOps:** A utilização de ferramentas como Azure DevOps para gerenciamento do backlog e controle de versão foi reconhecida como eficiente para o planejamento e acompanhamento das atividades.

3.4.3. Principais Problemas Identificados

Por outro lado, os entrevistados também apontaram desafios e problemas significativos que afetavam o processo de desenvolvimento:

- **Complexidade na Especificação de Requisitos:** Os critérios de aceitação e outros requisitos eram frequentemente descritos de maneira extensa e não diretamente aplicável aos desenvolvedores, levando a interpretações equivocadas e retrabalho.
- **Falta de Exemplos nos Protótipos:** A ausência de exemplos de mensagens de erro e interações com o usuário nos protótipos dificultava o trabalho dos desenvolvedores durante a fase de codificação.
- **Informações Fragmentadas em Diferentes Artefatos:** A dispersão de informações em múltiplos artefatos, como documentos de arquitetura e comentários técnicos, dificultava a localização e entendimento completo dos requisitos pelos desenvolvedores.

- Especificação de Requisitos Desatualizada: A prática de atualizar a especificação de requisitos simultaneamente à codificação, sem uma gestão adequada, resultava em histórias inconsistentes e sujeitas a alterações frequentes.

3.4.4. Oportunidades de Melhoria Identificadas

Com base nas entrevistas, foram identificadas várias oportunidades para melhorar o processo de desenvolvimento:

- Melhoria na Escrita de Critérios de Aceitação: Há uma necessidade clara de simplificar e objetivar os critérios de aceitação para evitar mal-entendidos e reduzir o tempo gasto na interpretação.
- Consolidação e Padronização dos Artefatos de Requisitos: Consolidar informações dispersas em um único documento ou ferramenta poderia melhorar a acessibilidade e compreensão dos requisitos por toda a equipe.
- Implementação de Exemplos nos Protótipos: Incluir exemplos concretos de interações com usuários nos protótipos poderia agilizar o desenvolvimento e melhorar a precisão das funcionalidades implementadas.
- Adoção de Ferramentas Especializadas em Gestão de Requisitos: Investir em uma ferramenta especializada para especificação e gestão de requisitos poderia resolver problemas associados à atualização simultânea e falta de centralização das informações.

3.5. Considerações Finais

Esta fase culmina na identificação de oportunidades de melhoria nos processos de especificação de requisitos e no alinhamento das expectativas quanto à implementação da abordagem RSD. A análise detalhada da situação atual é essencial para garantir que a transição para a abordagem RSD fosse realizada de maneira efetiva e orientada para os resultados.

4. IMPLANTAÇÃO DA ABORDAGEM RSD

Neste capítulo detalhamos a fase de implantação da abordagem RSD em um projeto de desenvolvimento ágil selecionado para estudo. Descreve a capacitação da equipe, a adaptação dos processos existentes, e o monitoramento durante a aplicação da abordagem, visando uma integração com as atividades de desenvolvimento da empresa.

4.1. Capacitação dos Envolvidos na Pesquisa e Aplicação da Abordagem

Antes da implementação da abordagem no âmbito do projeto, foi conduzido um treinamento por meio de videoconferência com o propósito de introduzir a equipe à abordagem RSD. A capacitação, que teve a duração de 4 horas divididas em 2 dias, foi organizada da seguinte maneira: No primeiro dia, o treinamento focou nos desenvolvedores, detalhando a estrutura da abordagem e como utilizá-la. No dia seguinte, gerentes e analistas foram instruídos, recebendo uma visão geral da estrutura e compreensão do que seria entregue aos desenvolvedores por meio dela.

Um total de 14 pessoas participaram do treinamento, abrangendo desenvolvedores, gerentes e analistas. Esta diversidade de participantes garantiu uma compreensão abrangente do novo conjunto de diretrizes por toda a equipe envolvida.

Detalhes acerca das práticas inerentes à abordagem foram minuciosamente elucidados, fornecendo uma compreensão abrangente. Em seguida, um piloto foi apresentado com alguns requisitos especificados usando a abordagem, esclarecendo dúvidas e familiarizando a equipe com a nova metodologia.

Após a capacitação, e considerando a natureza contínua do projeto, procedeu-se à criação integral de todos os modelos por meio da implementação da abordagem, viabilizando a sua incorporação na sprint subsequente. Como as atividades já estavam criadas no *Azure DevOps*, para diminuir o impacto de reestruturar a interface da ferramenta para a criação dos modelos, optou-se pela ferramenta *Google Sheets*, dada a sua facilidade de compartilhamento e baixa curva de aprendizado. Por ser uma ferramenta *online*, todos os membros da equipe podem acessar os modelos, que estão organizados em abas, facilitando a navegação, edição e adição de comentários a modelos específicos, conforme evidenciado na Figura 5.

Figura 5 – Modelo da abordagem criado no *Google Sheets*

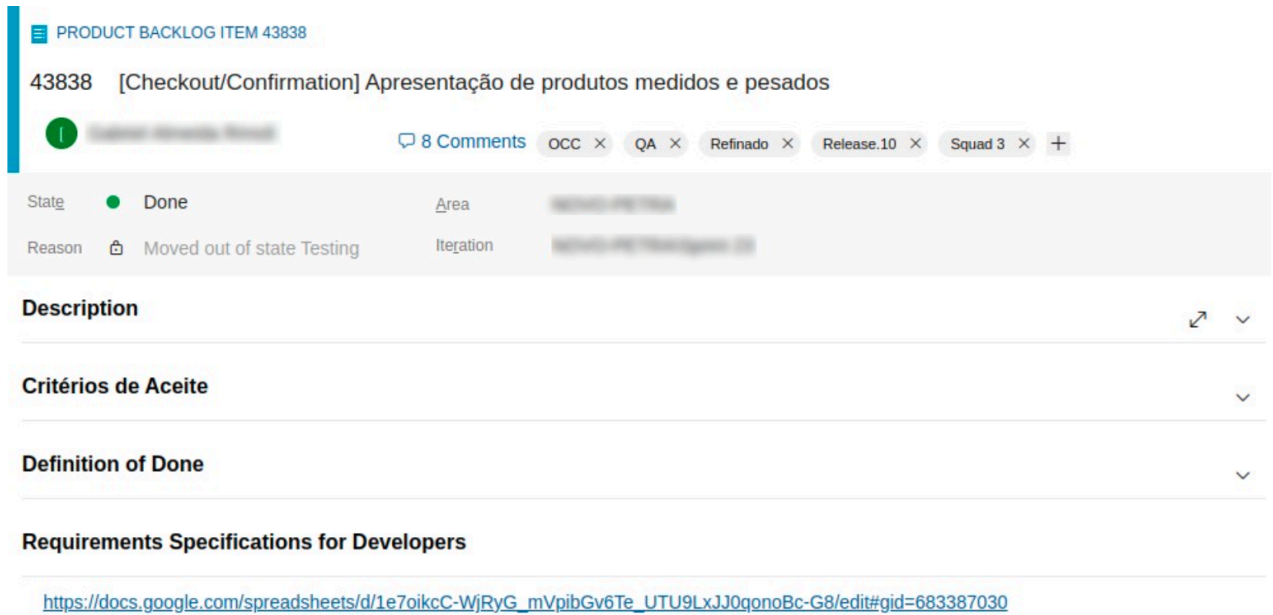
LABEL		SOURCE
Título da atividade		Responsável
DESCRIPTION		
Descrição da atividade		
DEPENDENCES		SOURCE
Link da atividade que está dependente, se houver		Responsável
INTERFACE	PAGES	SOURCE
Link da interface		Responsável
Imagem da interface		
ACCEPTANCE CRITERIA		
AC01	Critério de aceitação	
ADDITIONAL INFORMATION		SOURCE
Informações adicionais para o desenvolvimento		Responsável
SERVICES AND INTEGRATIONS		SOURCE
Link da documentação, swagger ou wiki		Responsável
USEFUL LINKS		SOURCE
Link úteis de documentações		Responsável

US 43748 ▾
Legendas ▾

Fonte: Adaptado de Medeiros (2017).

Para ter acesso aos modelos, os desenvolvedores acessavam o *link* adicionado a cada atividade no *Azure DevOps*, que redireciona para o *Google Sheets* com o RSD específico da atividade, como mostra a Figura 6.

Figura 6 – Campo adicionado ao *Azure DevOps* com *link* do RSD



4.1.1. Modelo de Capacitação

O treinamento para a equipe foi detalhado e estruturado para abordar todos os aspectos essenciais da abordagem RSD. Durante o primeiro dia, os desenvolvedores foram introduzidos à estrutura da abordagem, com ênfase em como utilizá-la efetivamente no desenvolvimento de projetos. Este módulo incluiu tópicos como a definição de requisitos, a utilização de modelos específicos e a integração da abordagem RSD nas práticas diárias. Para facilitar o entendimento da abordagem, foi utilizado o modelo criado no *Google Sheets* (Figura 5) apresentado na seção anterior. No segundo dia, o foco foi voltado para gerentes e analistas, fornecendo uma visão geral da abordagem e esclarecendo como os desenvolvedores iriam aplicar os conceitos aprendidos. Trechos dos slides usados no treinamento, como exemplos de requisitos e fluxos de trabalho, foram apresentados para facilitar a compreensão.

O treinamento foi dividido em duas sessões de 2 horas cada, ao longo de dois dias, para permitir uma absorção gradual do conteúdo. A metodologia incluiu apresentações expositivas, seguidas de dinâmicas de grupo onde os participantes puderam discutir e questionar os conceitos apresentados. Estudos de caso específicos foram utilizados para demonstrar a aplicação prática da abordagem RSD, e exemplos práticos ajudaram a equipe a visualizar como a metodologia se integraria ao fluxo de trabalho existente. Sessões de perguntas e respostas foram realizadas ao final de cada dia para resolver quaisquer dúvidas.

Durante o treinamento, várias questões foram levantadas pela equipe, destacando as áreas que necessitavam de maior clareza. Entre as principais questões, estavam dúvidas sobre a integração da abordagem RSD com ferramentas já existentes e como lidar com requisitos incertos. Essas questões foram abordadas com exemplos práticos adicionais e esclarecimentos específicos,

garantindo que todos os participantes tivessem uma compreensão clara da aplicação da abordagem.

4.1.2. Dificuldades e Lições Aprendidas Durante a Capacitação

Uma das principais dificuldades encontradas durante a capacitação foi a preocupação inicial de alguns membros da equipe à adoção de uma nova metodologia, por se tratar de um projeto em andamento. Além disso, alguns conceitos da abordagem RSD foram considerados complexos e difíceis de assimilar rapidamente, especialmente para aqueles com menos experiência em metodologias ágeis.

Para superar essas dificuldades, foram realizados ajustes na metodologia de ensino. Foram incluídos mais exemplos práticos específicos ao contexto da empresa, o que facilitou a compreensão e aplicação dos conceitos. As sessões de perguntas e respostas foram ampliadas, permitindo que os participantes pudessem esclarecer suas dúvidas em maior detalhe. Também foram incorporadas dinâmicas interativas para tornar o aprendizado mais engajador e acessível.

Com base nas experiências desta capacitação, algumas lições importantes foram aprendidas para futuros treinamentos. A necessidade de incluir mais exemplos práticos e contextuais foi evidente, ajudando a equipe a visualizar a aplicação da metodologia em cenários reais. Além disso, a importância de sessões de perguntas e respostas mais frequentes foi destacada, permitindo uma abordagem mais personalizada e eficiente para resolver dúvidas. Finalmente, a inclusão de sessões de feedback durante o treinamento ajudou a adaptar o conteúdo de forma dinâmica, atendendo melhor às necessidades dos participantes.

4.2. Monitoramento da Utilização da Abordagem

O monitoramento da utilização da abordagem foi realizado por meio do acompanhamento individual e diário dos desenvolvedores, com o intuito de identificar como os requisitos estavam sendo utilizados e as dificuldades encontradas. Durante a fase de monitoramento, foram elaborados um total de 26 RSDs, que detalharam os diversos aspectos dos requisitos e práticas do projeto. Dentro desses documentos, foram especificados 122 critérios de aceitação, assegurando que cada requisito tivesse um conjunto claro e objetivo de condições que deveriam ser atendidas para a sua aprovação.

A fase de monitoramento se estendeu por três sprints, onde cada sprint durou aproximadamente 15 dias. No decorrer das sprints, cada desenvolvedor era acompanhado diariamente para identificar se havia alguma dificuldade. A observação constante permitiu identificar problemas rapidamente e fornecer suporte imediato, garantindo uma integração mais suave da abordagem RSD.

Após o encerramento do período de monitoramento da utilização da abordagem, foi dado início à segunda fase de coleta de dados baseada em novas entrevistas semiestruturadas para

coletar a visão dos desenvolvedores em relação às melhores práticas, dificuldades e limitações da abordagem proposta. Cada entrevista teve uma média de 1 hora, e uma semana foi necessária para concluí-las, tendo em vista a disponibilidade da equipe. Todas as entrevistas foram realizadas por videochamadas, utilizando a ferramenta Teams. Assim como na primeira fase, todas as entrevistas foram gravadas e em seguida transcritas.

As entrevistas revelaram que, embora a maioria dos desenvolvedores tenha se adaptado bem aos critérios de aceitação, alguns sugeriram melhorias em áreas específicas, destacando a necessidade de ajustes contínuos para melhor atender as necessidades do projeto. No geral, a fase de monitoramento permitiu coletar dados valiosos para a melhoria contínua da aplicação da abordagem RSD.

4.3. Dificuldades Identificadas pelo Pesquisador no Processo de Implantação

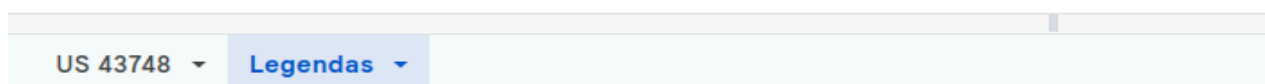
Por se tratar de um projeto em andamento, umas das principais dificuldades foi a indisponibilidade de informações durante a construção dos artefatos para o time *Frontend*, tendo em vista que alguns serviços que seriam utilizados pelas interfaces estavam sendo desenvolvidos paralelamente pelo *Backend*. Além disso, por se tratar de um projeto com várias tecnologias e *squads*, um RSD possuía várias dependências, recursos externos e responsáveis diferentes.

Tendo em vista que adaptações do modelo estão previstas pela metodologia, para assegurar a flexibilidade e adaptabilidade da abordagem, algumas adaptações foram realizadas no modelo para atender às necessidades do projeto em análise.

Adicionalmente, foi inserida uma aba específica para legendas de cores das abas do modelo, facilitando a identificação dos artefatos que necessitam de atuação específica ou que estão prontos para desenvolvimento, como exibido na Figura 7.

Figura 7 – Legendas de cores das abas do modelo

	COMPLETA	<i>A US está pronta para desenvolvimento</i>
	FALTA INTERFACE, SERVIÇO OU MATERIAL DE APOIO	<i>A US possui dependência de um artefato de desenvolvimento e as tasks podem transbordar</i>
	FALTA INTERFACE, SERVIÇO E MATERIAL DE APOIO	<i>A US possui dependência de dois ou mais artefatos e as tasks não podem ser iniciada</i>



Vale destacar, que o status da legenda não impede o início do desenvolvimento, mas sinaliza o risco de atrasos na entrega devido à falta de informações.

Além das legendas, foram adicionados novos campos à estrutura dos RSDs para identificar as dependências (*dependences*), interfaces (*interface*), informações adicionais (*additional information*), serviços (*services and integrations*), links úteis (*useful links*), e seus respectivos responsáveis (*source*), como mostra o exemplo nas Figura 8 e Figura 9.

Figura 8 – Novos campos para dependências, interfaces e responsáveis

LABEL		SOURCE
[Checkout/Confirmation] Apresentação de produtos medidos e pesados		Wendell Almeida
DESCRIPTION		
Vendedor poderá visualizar o estoque dos produtos medidos e pesados conforme sua unidade de medida para poder atender a necessidade de quantidade do item do cliente.		
DEPENDENCES		SOURCE
https://dev.azure.com/.../workitems/edit/33989		Wendell Almeida
INTERFACE	PAGES	SOURCE
https://xd.adobe.com/view/...	14 e 15	Wendell Almeida

Figura 9 – Novos campos para informações adicionais, serviços e links úteis

ADDITIONAL INFORMATION	SOURCE
Conceito de medido e pesado: todo produto que é cobrado com unidade de medida diferente da unidade de medida solicitada pelo cliente: (Ex: cliente pede 5m de cobre, e nós cobramos o equivalente em quilos)	Wendell Almeida de Luz
Unidade de medida (Unidades, metros, KG..) altera conforme campo "unidade de medida" cadastrado no cadastro de produto no AX.	Wendell Almeida
SERVICES AND INTEGRATIONS	SOURCE
https://api-...com.br/mdm-bff-inventory-price/v1/cart	Wendell Almeida
USEFUL LINKS	SOURCE
https://docs.google.com/spreadsheets...	Wendell Almeida
https://docs.oracle.com/en/cloud/saas/cx-commerce/uoccs/create-and-edit-product-types.html#GUID-61BA2CBC-E0C3-4CA7-B3F3-E449491D6AD4	Wendell Almeida

Os novos campos foram inseridos para integrar as informações e facilitar a comunicação entre desenvolvedores e responsáveis por cada recurso.

4.4. Considerações Finais

As experiências, desafios e sucessos encontrados durante a implantação da abordagem descritos neste capítulo podem fornecer evidências valiosas para outras pesquisas futuras que tenham por objetivo implantar a abordagem RSD.

5. AVALIAÇÃO DA ABORDAGEM RSD

Este capítulo apresenta a avaliação da abordagem RSD, adotando um protocolo de pesquisa-ação criterioso como estratégia para responder à questão definida anteriormente na seção 1.2: “Como a abordagem RSD é avaliada pelos stakeholders durante o desenvolvimento ágil de um software?”. Com o objetivo de compreender integralmente seu impacto no processo de desenvolvimento ágil, utilizamos métodos para avaliar a abordagem, proporcionando um entendimento sobre suas vantagens, desafios e áreas para melhorias. Além disso, apresentamos neste capítulo as discussões e ameaças à validade.

5.1. Resultados da Aplicação do Roteiro das Entrevistas

A análise da percepção da equipe foi realizada a partir de duas categorias: “Conteúdo” e “Estrutura”, seguindo os mesmos critérios adotados pelo estudo realizado por Medeiros (2017). Nessa seção será detalhado e discutidos os dados obtidos através das entrevistas e observações.

5.1.1. Conteúdo

Durante a aplicação da abordagem, os desenvolvedores realizaram uma avaliação metódica de todos os 26 documentos de especificação produzidos. A metodologia de avaliação adotada estruturou-se em torno de 5 dimensões, obtidas a partir da análise de casos cruzados de seis estudos de caso industriais realizado por Medeiros et. al (2018), cujo significado é apresentado a seguir:

- **Objetividade:** Refere-se à clareza e precisão com que os requisitos são definidos. Um RSD objetivo contém informações claras e diretas, necessárias para a implementação, sem ambiguidades ou informações redundantes que possam confundir ou desviar o desenvolvedor do que é necessário.
- **Legibilidade:** Esta dimensão avalia quão facilmente os desenvolvedores podem entender e interpretar os RSDs. A alta legibilidade é alcançada através de uma apresentação clara das informações, organização lógica e uso de uma linguagem acessível, permitindo que os desenvolvedores compreendam rapidamente os requisitos.
- **Completeness:** Mede se os RSDs fornecem todas as informações necessárias para que os desenvolvedores possam progredir no desenvolvimento sem necessidade de informações adicionais. Isso inclui detalhes sobre funcionalidades, comportamentos esperados e dependências.
- **Volatilidade:** Indica a frequência e o volume de alterações que um RSD sofre ao longo do tempo. A volatilidade pode refletir a adaptação a novas descobertas ou

mudanças nos requisitos do projeto, sendo um indicativo da necessidade de refinamento dos RSDs para atender às expectativas dos stakeholders.

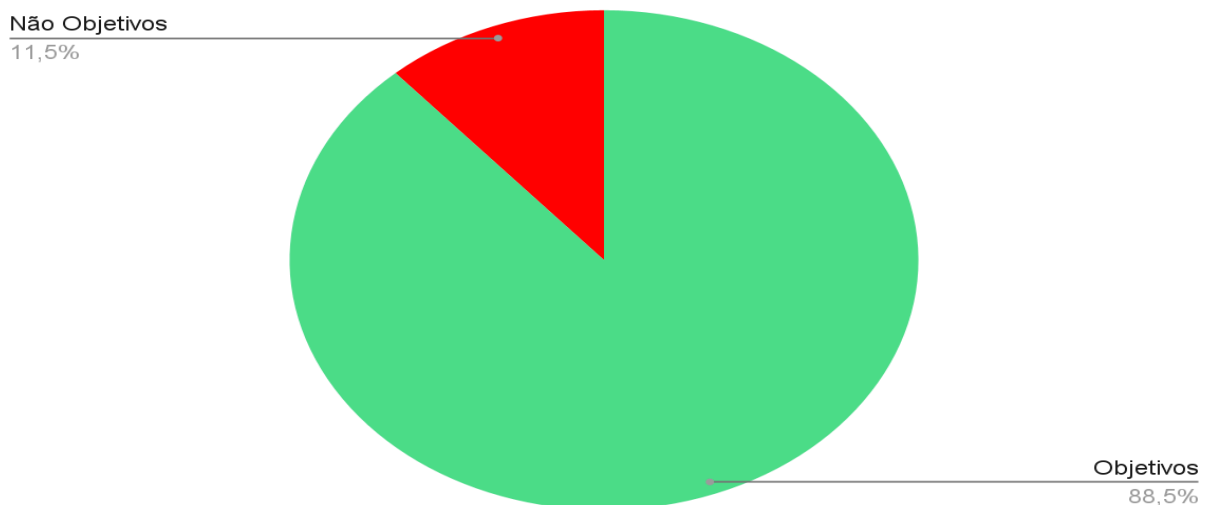
- **Conformidade:** Avalia até que ponto os RSDs estão alinhados com os padrões e práticas da equipe de desenvolvimento, assegurando que os artefatos sejam não apenas tecnicamente corretos, mas também úteis e aplicáveis no contexto do projeto.

Essas dimensões foram avaliadas pelos desenvolvedores por meio de entrevistas semiestruturadas, conforme apresentado no Apêndice B. As perguntas do questionário foram desenhadas para avaliar diretamente cada uma dessas dimensões, utilizando tanto escalas quantitativas, como a de 1 (inadequado) a 5 (muito adequado), quanto questões abertas que permitiram uma análise mais qualitativa. Por exemplo, a objetividade e a legibilidade foram medidas através de perguntas que pediam aos desenvolvedores para avaliar a estrutura do RSD e sua conformidade com critérios de qualidade específicos. A escala de 1 a 5 permitiu uma avaliação detalhada, enquanto as questões abertas possibilitaram aos participantes expressarem suas percepções de maneira mais abrangente.

A análise revelou que 88,5% dos RSDs, correspondentes a 23 artefatos, atenderam ao critério de objetividade, como mostra a Figura 10, fornecendo informações claras e diretas necessárias para a implementação. Contudo, 3 RSDs foram marcados como não objetivos, primariamente devido à inclusão de informações redundantes que não agregam valor ao desenvolvedor. Um caso notável foi onde um desenvolvedor implementou uma funcionalidade não prevista, evidenciando a importância da precisão na especificação dos requisitos.

Figura 10 – Percentuais de RSDs avaliados no critério de objetividade

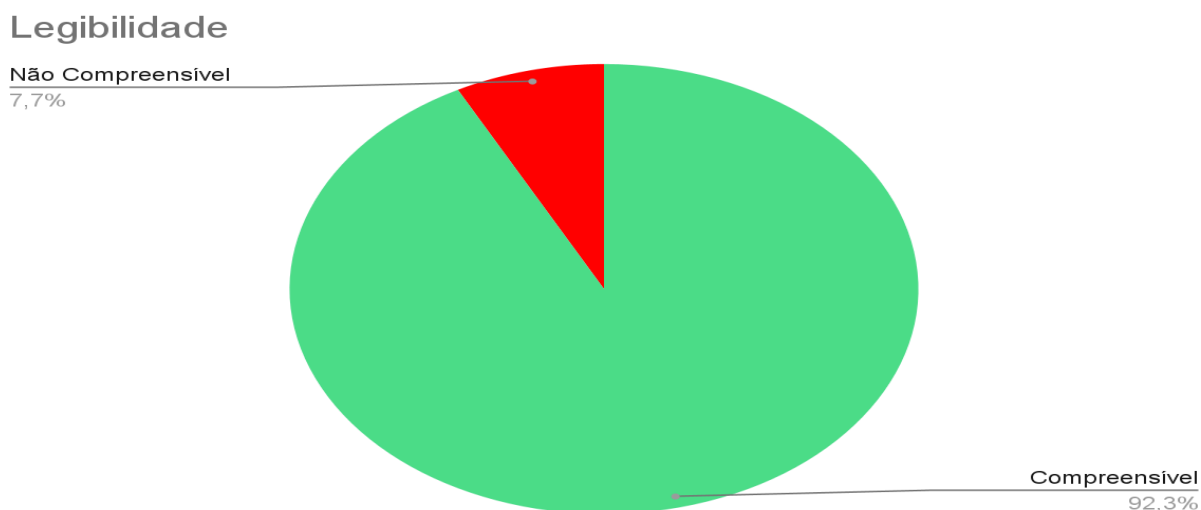
Objetividade



Em termos de legibilidade, 92,3% dos RSDs, correspondentes a 24 artefatos, foram considerados facilmente compreensíveis, como mostra a Figura 11, indicando uma apresentação

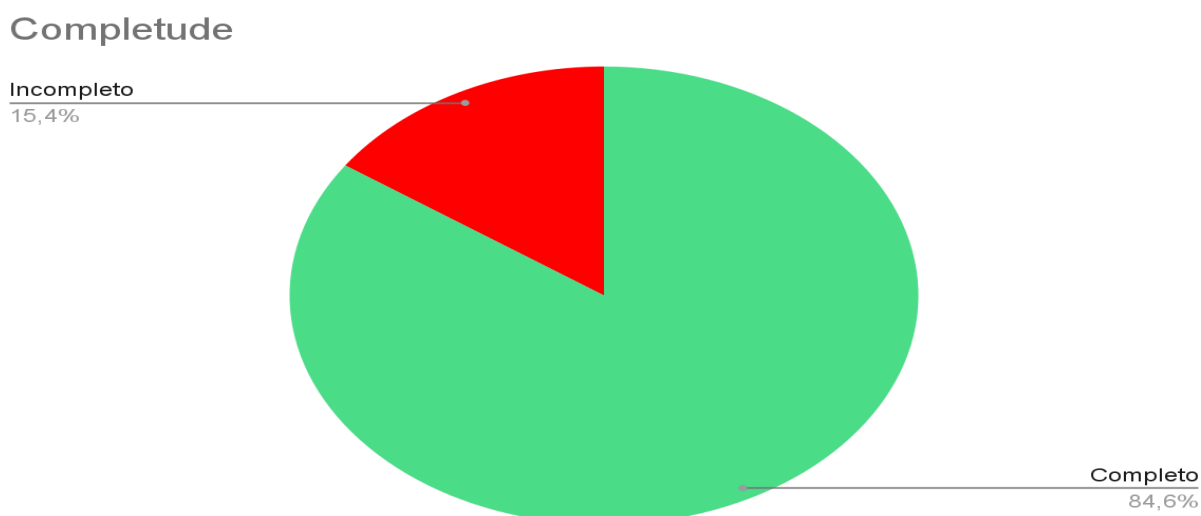
clara das informações. No entanto, a legibilidade de dois RSDs foi afetada negativamente por conterem mais de um requisito em um único critério de aceitação, complicando a interpretação por parte dos desenvolvedores.

Figura 11 – Percentuais de RSDs avaliados no critério de legibilidade



Quanto à completude, 84,6% dos RSDs, correspondentes a 22 artefatos, foram avaliados como suficientemente detalhados para permitir o desenvolvimento sem consultas adicionais, como mostra a Figura 12. A completude de quatro RSDs foi questionada devido à falta de informações críticas para a codificação, particularmente em casos em que os serviços associados ainda estavam em desenvolvimento. A necessidade de informações adicionais ou a espera pela conclusão de serviços relacionados foi citada como um desafio para a produtividade da equipe.

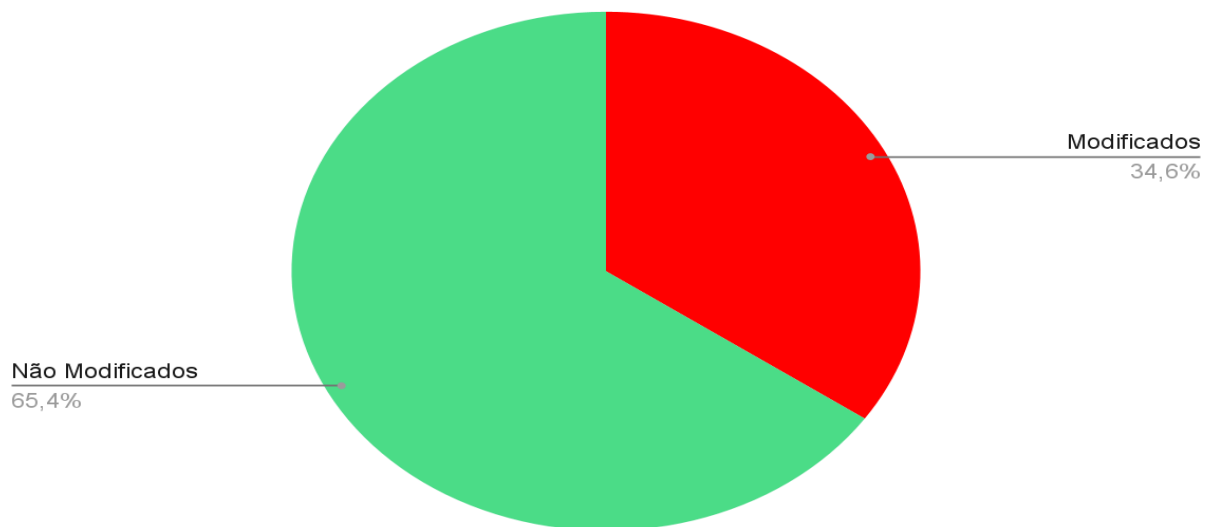
Figura 12 – Percentuais de RSDs avaliados no critério de completude



A análise da volatilidade, que mede as alterações feitas nos RSDs para aprimorá-los ou corrigi-los, identificou um total de 16 modificações. Vale ressaltar que um maior número de alterações não necessariamente indica uma melhoria na completude do RSD. Vale destacar, que 9 foram modificações sem duplicação, sendo assim 34,6% dos artefatos, como mostra a Figura 13. Essa taxa indica que, enquanto a maioria dos RSDs foi considerada adequada desde a sua concepção inicial, uma parcela significativa necessitou de ajustes. Isso pode ser visto tanto como uma oportunidade de aprendizado para refinar processos de elicitação e documentação de requisitos, quanto uma demonstração da flexibilidade da equipe em adaptar-se para garantir que os RSDs finais estejam alinhados com as necessidades do projeto e dos stakeholders.

Figura 13 – Percentuais de RSDs avaliados no critério de volatilidade

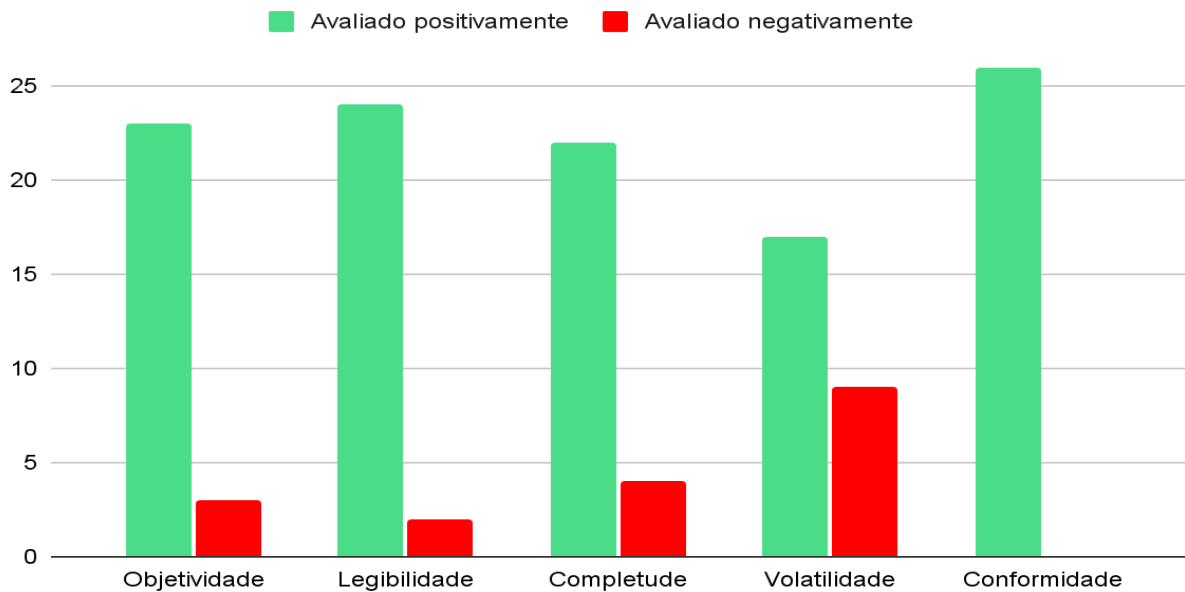
Volatilidade



De maneira encorajadora, todos os RSDs foram avaliados como estando em conformidade com o fator orientado à equipe, refletindo a adequação dos artefatos às necessidades e processos da equipe de desenvolvimento.

A Figura 14, consolida as informações apresentadas, permitindo uma visualização clara dos resultados obtidos, o que reforça a análise positiva através dos dados obtidos nas entrevistas.

Figura 14 – Percentuais de RSDs avaliados positivamente e negativamente



Esses resultados foram obtidos através de entrevistas com sete desenvolvedores de software, que compararam os RSDs com os Requisitos de Software do Sistema (SRSs) utilizados em projetos anteriores. A preferência clara pelos RSDs sobre abordagens anteriores ressalta sua superioridade em termos de completude, objetividade e legibilidade, oferecendo uma estrutura mais eficaz para o desenvolvimento de software.

5.1.2. Estrutura

Os desenvolvedores de software também foram entrevistados quanto à avaliação da estrutura dos RSDs com base em protótipos, critérios de aceitação e conceitos. Posteriormente, classificaram o RSD em uma escala de 1 (Inadequado) a 5 (Muito Adequado). De maneira geral, todos os entrevistados consideraram a estrutura do RSD como Muito Adequada. A equipe foi então indagada sobre os fatores de qualidade (Simplicidade, Informações Consolidadas, Informações técnicas, Critérios de Aceitação e Histórico de alterações), e todos os desenvolvedores enfatizaram que a estrutura estava alinhada com os fatores.

Um dos desenvolvedores ressaltou a importância das informações técnicas, sugerindo que a documentação de serviços deve ser consistentemente adicionada à atividade:

"Embora a atividade preveja a utilização de um serviço externo, é bastante comum a ausência de um documento auxiliar para orientar a utilização do serviço."

Conforme mencionado por Medeiros (2017), a abordagem RSD não impõe a adoção de uma ferramenta específica para a especificação de critérios de aceitação, construção de protótipos e o modelo conceitual. Cada empresa tem a liberdade de escolher as ferramentas de acordo com

suas necessidades, processos e metodologias. A equipe de desenvolvimento relatou que não enfrentou problemas ao utilizar o *Adobe XD* para a criação de protótipos e o *Google Sheets* para armazenar e elaborar o RSD e avaliou como um ponto positivo a flexibilidade da estrutura da abordagem para se adequar a necessidade real do projeto.

Por fim, a estrutura do RSD foi comparada com outras abordagens previamente utilizadas pelos desenvolvedores. Todos os entrevistados afirmaram que a estrutura do RSD é mais apropriada do que aquela de outras abordagens. Na avaliação comparativa por fatores de qualidade, todos consideraram que a estrutura do RSD é superior em termos de simplicidade, critérios de aceitação, aspectos técnicos e consolidação de informações.

Os resultados das entrevistas indicaram que o RSD atendeu às expectativas dos desenvolvedores de software em relação à sua estrutura. Alguns comentários dos entrevistados incluem:

"A descrição dos requisitos por meio de critérios de aceitação mais técnicos resulta em um SRS voltado para o desenvolvedor, fornecendo informações sobre como implementar o requisito. Isso nos auxilia na fase de codificação."

"A estrutura do RSD proporciona uma visão mais abrangente das regras e restrições aplicadas a um requisito específico."

5.2. Comparação da abordagem RSD em relação ao estado anterior

Seguindo os critérios utilizados pelo estudo realizado por Medeiros (2017), a avaliação da abordagem RSD em comparação com o estado anterior foi conduzida examinando como o trabalho da equipe foi influenciado, tendo contemplado os seguintes parâmetros: Esforço Necessário para Compreender, Codificar e Testar; Esforço Necessário para Estruturação; Análise de Impacto de Pedidos de Mudança e, por último, a Transferência de Conhecimento, Colaboração do Cliente e Dependência Entre as Partes Interessadas.

5.2.1. Esforço Necessário para Compreender, Codificar e Testar

A consolidação das informações proporcionou uma visão abrangente do contexto a ser desenvolvido. Como resultado, a equipe de desenvolvimento reduziu a necessidade de recorrer ao cliente ou a outra equipe para compreender comportamentos, regras e objetivos. Alguns desenvolvedores enfatizaram que a consolidação das informações teve um impacto positivo em sua produtividade:

"A consolidação de informações facilita nossa compreensão do todo, diminuindo a dependência de consultar outras pessoas para entender"

critérios específicos que não estavam inicialmente descritos na atividade, mas sim em uma segunda ferramenta."

"Agora, não é mais necessário acessar três ferramentas diferentes em busca de informações não claras. O RSD consolida as informações técnicas necessárias para o desenvolvimento."

Na visão dos desenvolvedores, a descrição dos critérios de aceitação com enfoque técnico também contribuiu para a redução do esforço na codificação dos requisitos.

Anteriormente, os testes de aceitação eram conduzidos com base na descrição da história do usuário e nos critérios de aceitação voltados para o cliente. A adoção da abordagem RSD sistematizou os testes a partir dos critérios de aceitação, resultando em uma diminuição significativa do esforço necessário.

5.2.2. Esforço Necessário para Estruturação

Levando em conta a curva de aprendizagem inerente à adoção de novas práticas, a estruturação do RSD revelou-se um processo que demandou um investimento de tempo considerável, excedendo o esforço usualmente dedicado à elaboração de histórias de usuários. Isso foi destacado por um membro da equipe, que observou:

"A estrutura exige um esforço maior para a descrição de critérios voltados para os desenvolvedores e a consolidação das informações técnicas e de negócios."

No entanto, a equipe destacou que esse esforço inicial foi crucial para estabelecer uma base sólida:

"A estruturação detalhada e consolidada do RSD permitiu uma compreensão mais profunda dos requisitos, facilitando o entendimento e a codificação posterior pelo time de desenvolvimento."

Segundo a equipe, a flexibilidade na adaptação da estrutura e a variedade de ferramentas disponíveis também foram destacadas como pontos positivos, possibilitando uma abordagem personalizada e eficaz na construção do RSD.

5.2.3. Análise de Impacto de Pedidos de Mudança

Após a condução da avaliação de impacto das mudanças pela equipe de apoio, observou-se que a capacidade de resposta do projeto às solicitações de mudança melhorou significativamente. A priorização ajustada dos requisitos e critérios de aceitação, aliada a uma compreensão mais profunda dos impactos dessas mudanças, permitiu uma gestão mais eficaz do escopo do projeto. Como resultado, a equipe conseguiu não apenas atender às demandas de mudança de forma mais

ágil, mas também otimizar o planejamento e alocação de recursos, minimizando interrupções no fluxo de trabalho e promovendo uma entrega mais contínua de valor. Este processo reforçou a importância de uma gestão de mudanças eficiente e destacou o papel crítico da equipe de apoio no sucesso do projeto ágil.

5.2.4. *Transferência de Conhecimento, Colaboração do Cliente e Dependência Entre as Partes Interessadas*

Embora os critérios de aceitação do tipo técnico tenham diminuído a dependência das partes interessadas e o esforço necessário para codificar um requisito, na visão dos desenvolvedores, as reuniões diárias continuam sendo fundamentais para esclarecer dúvidas relacionadas aos critérios de aceitação do tipo de negócio e para compartilhar desafios com outros membros da equipe.

Segundo a equipe, a comunicação com o cliente estava em um nível adequado. Todos os membros da equipe mantinham interações com o cliente sempre que necessário, pelo menos uma vez por semana. As comunicações mais frequentes eram conduzidas por meio de mensagens ou vídeo chamadas utilizando o aplicativo *Teams*. O RSD não era utilizado para validar requisitos com os clientes; em vez disso, as validações ocorreram por meio das histórias de usuário escritas pelo cliente e das entregas frequentes de software.

5.3. Resultados da Avaliação dos Critérios de Pesquisa-Ação

Conforme detalhado na seção 3.2.3.3, foram definidos 3 critérios de avaliação da pesquisa, onde cada critério foi avaliado com base em sua eficácia em promover entendimento, gerar e assimilar conhecimento, e contribuir para o campo do desenvolvimento ágil de software. Abaixo estão os principais resultados:

- Primeiro critério - *Facilitação do Entendimento Comum do Projeto*: Os resultados desta avaliação indicaram que a aplicação da abordagem RSD foi eficaz em promover um entendimento compartilhado do projeto entre todos os envolvidos. As entrevistas e observações diretas destacaram a clareza dos objetivos do projeto, a eficácia da comunicação interna e a compreensão dos requisitos especificados pela abordagem RSD.
- Segundo critério - *Geração e Assimilação de Novo Conhecimento*: A pesquisa demonstrou que o RSD foi capaz de gerar novo conhecimento de maneira eficaz, facilitando sua assimilação pelos envolvidos. Isso foi evidenciado pela produção de artefatos práticos e pela aplicabilidade das soluções desenvolvidas no contexto do projeto, além da reflexão sobre os aprendizados ao longo do processo.
- Terceiro critério - *Contribuição para o Desenvolvimento Ágil de Software*: A avaliação revelou que a abordagem RSD contribuiu significativamente para o

campo do desenvolvimento ágil de software, especialmente em relação à especificação de requisitos. Os participantes destacaram a utilidade e aplicabilidade da abordagem RSD, assim como a aderência das práticas implementadas com as melhores práticas da indústria. Essa percepção foi obtida principalmente através das respostas às perguntas Q12 e Q17 do roteiro de entrevistas (Apêndice B), que questionavam sobre as práticas que melhoraram o desempenho do projeto e as boas práticas que poderiam ser aplicadas em outros contextos.

É importante destacar que os resultados obtidos reforçam a contribuição da abordagem RSD em diferentes aspectos do desenvolvimento ágil de software. A partir das avaliações detalhadas dos critérios estabelecidos, ficou evidente que o RSD não apenas facilitou um entendimento comum do projeto entre os envolvidos, mas também promoveu a geração e assimilação de novo conhecimento de maneira eficaz. Além disso, sua contribuição para a especificação de requisitos foi positivamente reconhecida pelos participantes do estudo, indicando sua relevância e aplicabilidade prática.

5.4. Síntese Temática dos Resultados Após Aplicação da Abordagem

Para fornecer uma visão das percepções coletadas durante a pesquisa-ação, realizou-se uma síntese temática dos dados coletados através das entrevistas semiestruturadas e observações diretas.

5.4.1. Processo de Síntese Temática

Seguindo o método preconizado por Cruzes e Dyba (2011), conforme descrito na seção 3.3, as etapas seguidas do processo foram as seguintes:

1. **Leitura Preliminar dos Dados:** A leitura preliminar das entrevistas permitiu identificar padrões nas respostas dos participantes, principalmente sobre suas percepções da abordagem RSD, suas facilidades, dificuldades e sugestões para melhorias.
2. **Identificação e Rotulação de Segmentos (Códigos):**
 - **Eficácia na Comunicação e Entendimento Compartilhado:**
 - "*Centralização das informações*": Entrevistado mencionou como a abordagem RSD centraliza informações essenciais, facilitando a localização dos dados necessários para o desenvolvimento.
 - "*Facilidade de uso*": Um entrevistado destacou que a abordagem RSD ajudou a ter uma visão mais completa das tarefas e dependências, melhorando a comunicação interna.
 - **Facilitação da Geração e Assimilação de Conhecimento:**

- "*Aprendizado fácil*": Entrevistado mencionou que o processo de aprendizagem da abordagem foi intuitivo e facilitou o entendimento das tarefas.
- "*Visão clara e organizada*": Entrevistado relatou que a organização do RSD permitiu uma visão clara das dependências, facilitando o trabalho e o aprendizado.
- **Impacto Positivo na Especificação de Requisitos:**
 - "*Objetividade e clareza*": Entrevistados valorizaram a clareza e a objetividade proporcionadas pela abordagem, mencionando como isso ajudou na codificação e no entendimento das tarefas.
 - "*Qualidade das informações*": Entrevistado elogiou a forma consolidada como as informações eram apresentadas no RSD, o que facilitava o desenvolvimento.
- **Desafios e Oportunidades de Melhoria:**
 - "*Complexidade na gestão de dependências*": Entrevistado mencionou desafios relacionados às dependências entre atividades, sugerindo que isso poderia ser melhorado.
 - "*Necessidade de mais detalhes técnicos*": Entrevistado sugeriu que a abordagem poderia incluir mais informações técnicas para melhorar a eficácia no desenvolvimento.

3. Agrupamento de Códigos em Temas:

- **Comunicação e Entendimento:** Inclui códigos relacionados à centralização das informações e à facilidade de uso que melhoraram a comunicação e o entendimento compartilhado.
- **Geração e Assimilação de Conhecimento:** Abrange códigos que destacam o aprendizado fácil e a visão clara proporcionada pela abordagem RSD.
- **Especificação de Requisitos:** Reflete códigos sobre a clareza, objetividade, e a qualidade das informações na especificação de requisitos.
- **Desafios e Melhorias:** Envolve códigos que indicam desafios como a gestão de dependências e a necessidade de mais detalhes técnicos.

4. **Identificação de Categorias Abrangentes:** Os temas emergentes foram organizados em categorias mais amplas para facilitar a compreensão e análise dos resultados. Os principais temas identificados:

- **Eficácia na Comunicação e Entendimento Compartilhado:** A abordagem RSD demonstrou ser eficaz na promoção de um entendimento comum do projeto, melhorando a comunicação interna e externa.
- **Facilitação da Geração e Assimilação de Conhecimento:** Os dados coletados evidenciaram que o RSD facilitou a geração de novo conhecimento, possibilitando sua assimilação de maneira mais eficiente pelos membros da equipe.
- **Impacto Positivo na Especificação de Requisitos:** Os participantes reconheceram o impacto positivo da abordagem RSD na especificação de requisitos, destacando a clareza e a objetividade alcançadas.
- **Desafios e Oportunidades de Melhoria:** Foram identificados desafios, como a complexidade na gestão de dependências entre atividades, e oportunidades de melhoria, como a necessidade de maior integração de ferramentas para melhorar a eficiência.

Esses temas e categorias oferecem uma visão estruturada dos resultados da pesquisa-ação, destacando tanto os benefícios percebidos quanto às áreas que podem ser aprimoradas na implementação da abordagem RSD.

5.5. Ameaças a Validade

No âmbito de projetos de pesquisa-ação, a presença de fatores não controlados representa uma ameaça à validade, podendo influenciar os resultados, especialmente devido a mudanças internas ou eventos inesperados. Para mitigar essa ameaça durante a implantação e monitoramento da abordagem RSD, adotou-se uma análise comparativa dos dados obtidos por meio das entrevistas. Utilizando perguntas relacionadas ao estado do projeto, busca-se isolar os efeitos da abordagem aplicada, identificando possíveis mudanças ocorridas durante a implementação e avaliando o impacto dessas mudanças no resultado final.

Outra ameaça considerável é o nível de experiência dos desenvolvedores, uma variável que pode exercer uma influência significativa na validade desta pesquisa. A percepção sobre a efetividade da abordagem RSD pode variar entre desenvolvedores com diferentes níveis de experiência, impactando os resultados. Para enfrentar essa ameaça, foi incorporada uma identificação do nível de experiência dos entrevistados durante as entrevistas. Essa abordagem possibilitará uma análise mais precisa, considerando a experiência individual de cada desenvolvedor. A análise dos resultados levou em conta a variação na experiência, proporcionando uma compreensão aprofundada do impacto da abordagem RSD em diferentes níveis de habilidade. Essa estratégia contribuirá para a robustez e validade dos resultados obtidos na pesquisa.

A sensibilidade dos métodos de coleta de dados também é uma ameaça potencial à validade da pesquisa. Para mitigar esse risco, optou-se por entrevistas semiestruturadas adaptadas ao contexto específico do projeto, sendo conduzidos testes-piloto para identificar possíveis

problemas no roteiro. Durante as entrevistas, enfatizou-se a necessidade de respostas detalhadas, evitando perguntas sugestivas. Além disso, foram utilizadas sondagens para explorar mais profundamente as respostas dos entrevistados, evitando respostas simples, vagas ou superficiais. Essa abordagem busca capturar argumentos, exemplos e detalhes que enriquecem a compreensão da percepção de cada desenvolvedor de software, garantindo a sensibilidade adequada na coleta de dados.

Por fim, a mudança da abordagem de especificação em um projeto já em andamento também é uma ameaça potencial. Essa mudança pode introduzir variáveis desconhecidas que afetam os resultados da pesquisa, como a resistência à mudança por parte da equipe, a necessidade de reestruturação de processos e a interrupção do fluxo de trabalho existente. Para mitigar essa ameaça, foi adotada uma abordagem gradual na implementação da abordagem RSD, envolvendo a equipe desde as fases iniciais de planejamento e fornecendo suporte contínuo durante a transição. Além disso, foram realizadas avaliações periódicas do progresso e do impacto da mudança, permitindo ajustes conforme necessário para garantir a continuidade e a eficácia do projeto. Essa abordagem proativa visa minimizar os riscos associados à mudança de abordagem e garantir a validade dos resultados da pesquisa.

Em conjunto, essas estratégias visam atenuar as ameaças à validade, fortalecendo a confiabilidade e solidez da pesquisa ao avaliar a efetividade da abordagem RSD em um contexto ágil

5.6. Considerações Finais

O resultado da avaliação ressalta a relevância da abordagem RSD no fortalecimento da prática de especificação de requisitos em ambientes ágeis. As evidências obtidas não apenas confirmam o valor prático da RSD, mas também abrem caminho para futuras investigações que possam expandir seu escopo e aplicabilidade. Encorajamos a exploração contínua deste campo, com ênfase na adaptação e refinamento da abordagem para atender a uma gama ainda mais ampla de contextos de desenvolvimento ágil. O próximo capítulo se concentrará nas considerações finais da pesquisa, consolidando os resultados e proporcionando uma visão geral das contribuições deste estudo.

6. CONCLUSÃO

O propósito deste estudo consistiu em avaliar a implementação da abordagem RSD na prática, com ênfase na identificação de suas vantagens e desvantagens no contexto da aplicação e utilização dessa abordagem. Não era nosso intento realizar uma comparação direta do RSD com outras abordagens, embora algumas indagações nesse sentido tenham sido contempladas nas entrevistas.

A aplicação da abordagem RSD em um ambiente real de projeto ágil ofereceu evidências valiosas sobre sua eficácia, desafios e impacto na especificação de requisitos. A pesquisa revelou que a implementação da abordagem RSD facilitou uma compreensão mais clara e detalhada dos requisitos do software pelas equipes de desenvolvimento, contribuindo para uma base mais sólida na construção do software. A utilização de protótipos e a especificação detalhada dos critérios de aceitação se mostraram particularmente eficazes para melhorar a comunicação entre os membros da equipe, facilitando a identificação integrada de requisitos funcionais, não funcionais e técnicos.

Por outro lado, a implementação da abordagem RSD também enfrentou desafios, como a necessidade de um maior investimento de tempo na fase inicial do projeto para a capacitação da equipe e a adaptação dos processos existentes. A complexidade adicional, introduzida pela necessidade de manter uma documentação mais detalhada, exigiu um esforço extra de gestão por parte da equipe de desenvolvimento. Esses resultados indicam que, embora a abordagem RSD seja avaliada positivamente pelos stakeholders em termos de sua contribuição para a clarificação e comunicação de requisitos, ela também exige uma consideração cuidadosa de fatores como o investimento inicial em tempo e recursos, além do gerenciamento de sua implementação no contexto das práticas e processos já estabelecidos pela equipe de desenvolvimento.

Os resultados obtidos indicam que o RSD atendeu às expectativas dos desenvolvedores, revelando-se uma abordagem objetiva e apropriada para as atividades de codificação. As práticas introduzidas não apresentaram impacto negativo na agilidade do processo de desenvolvimento. Os achados sustentam a ideia de que o RSD tem potencial para reduzir a lacuna entre os domínios do problema e da solução, permitindo que os desenvolvedores alcancem uma compreensão mais precisa da funcionalidade a ser implementada. Além disso, a abordagem RSD resulta em uma especificação de requisitos mais alinhada com a implementação pretendida, possibilitando a representação integrada de aspectos técnicos juntamente com os requisitos funcionais.

A comparação dos resultados desta pesquisa com o artigo "Requirements specification for developers in agile projects: Evaluation by two industrial case studies" (MEDEIROS et al., 2020) revelou semelhanças significativas e trouxe evidências valiosas para a aplicação da abordagem RSD em ambientes de desenvolvimento ágil. Ambos os estudos destacaram a eficácia da abordagem RSD em melhorar a clareza, objetividade e integração dos requisitos de software, contribuindo assim para uma especificação de requisitos mais eficiente e alinhada com as

necessidades do desenvolvimento. Especificamente, a integração de AC+s, protótipos e entidades de dados foi fundamental para a melhoria da legibilidade e compreensão dos artefatos RSD, um aspecto positivo observado tanto nesta dissertação quanto nos estudos de caso do artigo. A flexibilidade na adaptação da abordagem RSD ao processo de desenvolvimento foi igualmente reconhecida como um fator positivo, permitindo uma implementação eficaz ajustada às particularidades de cada projeto.

Porém, desafios comuns também foram identificados, como dificuldades em registrar os resultados dos testes de aceitação sem ferramentas especializadas e o suporte inadequado para rastrear AC+s. Estes aspectos apontam para a necessidade de ferramentas ou metodologias complementares que facilitem o gerenciamento e a reutilização eficiente dos AC+s. Os resultados desta pesquisa e da pesquisa realizada por Medeiros (2017) concordam que a adoção da abordagem RSD tem um impacto positivo geral sobre a produtividade e a qualidade do trabalho em equipe no desenvolvimento ágil, apesar de certos desafios operacionais que precisam ser resolvidos. Ambos os estudos enfatizam a importância da continuidade nas avaliações da abordagem RSD para explorar ainda mais seu potencial em diversos contextos de projeto, destacando a contribuição valiosa da abordagem para a Engenharia de Requisitos em Desenvolvimento Ágil de Software.

O feedback obtido por meio de entrevistas sugere que o RSD não implica em esforço adicional à equipe técnica, podendo, inclusive, contribuir para a redução do esforço envolvido em codificação, teste e manutenção. O desempenho dos testes a partir dos critérios de aceitação também foi destacado como um aspecto positivo. As avaliações desta aplicação não evidenciaram pontos de melhoria na estrutura da abordagem, mesmo quando comparada a outras metodologias.

No que diz respeito aos desafios enfrentados durante a implementação do RSD, foi observado que a produtividade do time funcional sofreu impacto quando uma atividade apresentava numerosas dependências de outras tarefas. Essas dependências resultaram na necessidade constante de atualização do RSD à medida que as interdependências eram concluídas. Adicionalmente, foi destacado que o processo de elaboração do RSD para uma atividade exigia mais tempo em comparação com as histórias de usuários previamente utilizadas pelo projeto. No entanto, a equipe enfatizou que esse investimento de tempo foi compensado pelo aumento da produtividade, graças à facilidade de acesso às informações essenciais para a codificação.

Destaca-se como ponto positivo da aplicação da abordagem a flexibilidade na adaptação da estrutura e o uso de diferentes ferramentas, tanto para a construção do RSD quanto para a criação de protótipos e modelos do projeto.

6.1. Contribuições da Pesquisa

Para respaldar a elaboração de requisitos no contexto do desenvolvimento ágil de software, este estudo apresenta as seguintes contribuições:

- **Estado da Arte:** Este estudo contribui para o estado da arte ao apresentar uma análise empírica detalhada da implementação prática da abordagem RSD em um ambiente ágil de desenvolvimento de software.
- **Protocolo de Pesquisa-Ação:** Descrevemos minuciosamente o desenho e procedimentos empregados para os estudos empíricos, fornecendo um protocolo de pesquisa-ação que serve como referência para futuras investigações em contextos similares.
- **Validação do RSD:** Os resultados indicam que o RSD atendeu às expectativas dos desenvolvedores, destacando-se como uma abordagem objetiva e adequada para atividades de codificação. A pesquisa valida a eficácia do RSD como uma ferramenta prática no processo de desenvolvimento ágil.
- **Propostas de Melhorias no RSD:** Apesar da eficácia geral, identificamos desafios, como a complexidade associada a atividades com muitas dependências, e áreas que demandam mais tempo durante a elaboração do RSD. Uma proposta de melhoria será apresentada na próxima seção de trabalhos futuros.
- **Publicações:** Os resultados da pesquisa foram detalhadamente documentados em um artigo atualmente em processo de revisão para submissão a uma conferência científica específica. Além disso, um resumo detalhado do protocolo de pesquisa foi apresentado em um simpósio de pesquisa, e submetido a uma conferência internacional, como mostra o item #1 e #2 da Tabela 3, que tem por objetivo sumarizar as publicações desta pesquisa até o momento.

Tabela 3 – Publicações desta pesquisa

	Conferências, Jornais, Revistas e Simpósios	Tipo	Conteúdo da publicação
#1	5º Simpósio de Pesquisa, Inovação e Pós-Graduação do Instituto Federal da Paraíba	Simpósio	Avaliação de uma Abordagem de Especificação de Requisitos: Definindo o Protocolo da Pesquisa-ação
#2	27º Conferência Internacional WER 2024	Conferência	Especificação de Requisitos em Projetos Ágeis: Avaliação de uma Abordagem Voltada para Equipes de Desenvolvimento
#3	The 40th ACM/SIGAPP Symposium On Applied Computing	Simpósio	Specification of Requirements in Agile Projects: Evaluation of an Approach Focused on Development Teams

6.2. Trabalhos Futuros

Durante a implementação da abordagem, observou-se que a produtividade do time funcional era prejudicada por atividades que apresentavam muitas dependências, conforme destacado nas seções anteriores. Isso resultava em atualizações frequentes dos artefatos. Além disso, o tempo necessário para a elaboração desses artefatos em uma atividade específica era significativamente maior em comparação com outras técnicas previamente utilizadas. Estes desafios específicos impulsionaram a busca por soluções e estratégias para atenuar esses impactos.

No que tange às oportunidades de aprimoramento do conhecimento atual sobre a atividade de especificação de requisitos por meio da abordagem RSD, planejamos dar continuidade a esta pesquisa através das seguintes iniciativas:

1. Desenvolver uma ferramenta web destinada a facilitar a implementação e criação dos RSDs, seguindo o modelo da abordagem criado no *Google Sheets* (Seção 4, Figura 5) adaptado do modelo de Medeiros (2017), incorporando as seguintes funcionalidades:
 - a. Acesso restrito a usuários com credenciais de acesso;
 - b. Integração com a plataforma *Azure DevOps* para extrair informações:
 - i. Número de identificação da atividade;
 - ii. Título da atividade;
 - iii. Descrição da atividade;
 - iv. Analista responsável pela atividade;
 - v. Desenvolvedor responsável pela atividade;
 - vi. Status;
 - vii. Critérios de aceitação;
 - viii. Atividades sucessoras e predecessoras;
 - ix. Comentários gerais;
 - c. Edição das informações integradas do *Azure DevOps*;
 - d. Repositório contendo critérios de aceitação gerais para promover a reutilização e padronização de componentes do sistema;
 - e. Busca de critérios de aceitação gerais no repositório;
 - f. Upload de interfaces no formato de imagem;
 - g. Adição de links de acesso rápido aos protótipos;
 - h. Status de completude;
 - i. Upload de documentações auxiliares;
 - j. Adição de links de acesso rápido as documentações externas;
 - k. Formatação dos critérios de aceitação em lista numerada para facilitar a leitura e identificação de critérios sucessores e predecessores;
 - l. Histórico de alterações do RSD;

2. Realizar uma avaliação abrangente da abordagem em um projeto ágil, empregando a ferramenta web que será desenvolvida para facilitar a implementação e criação dos RSDs. Nessa etapa, almejamos comparar os resultados obtidos com outras pesquisas que não fizeram uso da referida ferramenta, buscando identificar possíveis melhorias e diferenciais oferecidos pela ferramenta proposta.
3. Realizar estudo quantitativo para mensurar o impacto da abordagem RSD na produtividade das equipes de desenvolvimento, utilizando métricas específicas relacionadas ao tempo, esforço e qualidade do código produzido.

Essas sugestões visam explorar diferentes facetas da abordagem RSD e proporcionar uma compreensão mais abrangente de seu potencial e limitações no contexto de desenvolvimento de software.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, A. J. O planejamento de pesquisas qualitativas em educação. *Cadernos de pesquisa*, n. 77, p. 53-62, 1991.
- ALVES, E. et al. Specification and Management of Requirements in Agile Software Development: A Systematic Mapping Study. *Journal of Systems and Software*, v. 149, p. 53-79, 2019.
- BALLANTYNE, D. Dialogue and its role in the development of relationship specific knowledge. *Journal of Business & Industrial Marketing*, 2004.
- BECK, K. *Extreme Programming Explained: Embrace Change*. Reading, MA: Addison-Wesley Professional, 1999.
- BECK, K. *Test-Driven Development: By Example*. Addison-Wesley, 2003.
- BECK, Kent et al. *Manifesto for Agile Software Development*. 2001. Disponível em: <https://agilemanifesto.org/>. Acesso em: 08 maio 2023.
- BROWN, Judith et al. *Agile Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.
- CHEN, Y. et al. Collaboration Patterns and Knowledge Construction in Agile Software Development. *Journal of Systems and Software*, v. 85, n. 6, p. 1331-1346, 2012.
- COHN, M. *User Stories Applied: For Agile Software Development*. Upper Saddle River, NJ: Addison-Wesley Professional, 2004.
- COUGHLAN, P.; COUGHLAN, D. Action research for operations management. *International Journal of Operations & Production Management*, v. 22, n. 2, p. 220-240, 2002.
- CRUZES, D.S., DYBA, T. Recommended Steps for Thematic Synthesis in Software Engineering. *International Symposium on Empirical Software Engineering and Measurement*. Banff, AB, 2011, pp. 275-284. doi:10.1109/ESEM.2011.36, 2011.
- ACEITUNA, D., DO, H., WALIA, G. S., LEE, S.-W. Evaluating the use of model-based requirements verification method: A feasibility study. *Workshop on Empirical Requirements Engineering (EmpiRE 2011)*. Trento, Italy, 2011, pp. 13-20. doi:10.1109/EmpiRE.2011.6046248.
- DeLUCA, C., COAD, P. *Feature Driven Development: A practical guide*. Upper Saddle River, NJ: Prentice Hall, 1999.
- GOLDENBERG, M. *A arte de pesquisar: como fazer pesquisa qualitativa em Ciências Sociais*. Rio de Janeiro: Record, 1999.
- HIRSCHHEIM, R. *Information systems development: Challenges in practice, theory and education*. New York: Wiley, 1995.
- HUSSAIN, Z.; MALIK, S. E. Agile Requirements Engineering Practices and Challenges: A Grounded Theory Approach. *Journal of Systems and Software*, v. 107, p. 132-151, 2015.

- INAYAT, I.; NAWAZ, M. R. Effective Requirements Elicitation: Role of Prototyping and Mockups. *International Journal of Computer Science and Network Security*, v. 15, n. 12, p. 43-47, 2015.
- LOSADA, B., URRETAVIZCAYA, M., FERNÁNDEZ-CASTRO, I. A guide to agile development of interactive software with a User Objectives-driven methodology. *Science of Computer Programming*, Volume 78, Issue 11, 1. Pages 2268-2281, 2013.
- LOUCOPOULOS, P. Conceptual modeling: current state and future perspectives. In: JACOBSON, I.; WANDEL, J.; LIEBERHERR, K. (Eds.). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Boston: Addison-Wesley, 1992. p. 41-60.
- MACHADO, C.; ABREU, A.; SCHELP, E. *Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações*. Rio de Janeiro: Elsevier, 2018.
- MANIFESTO for Agile Software Development. 2001. Disponível em <http://www.agilemanifesto.org>. Acesso em: 16-Out-2021.
- MEDEIROS et al. Quality of software requirements specification in agile projects: A cross-case analysis of six companies. *Journal of Systems and Software*, v. 142, p. 171-194, 2018.
- MEDEIROS et al. Requirements specification for developers in agile projects: Evaluation by two industrial case studies. *Information and Software Technology*, v. 117, p. 106194, 2020.
- MEDEIROS, J. D. R. V. D. An approach to support the requirements specification in agile software development. Masters Dissertation, Universidade Federal de Pernambuco. March, 2017.
- MELLO, C. H. P.; TURRIONI, J. B.; XAVIER, A. F.; CAMPOS, D. F. Pesquisa-ação na engenharia de produção: proposta de estruturação para sua condução. *Production*, v. 22, p. 1-13, 2012.
- NEBE, J.; et al. Understanding the Integration of Mockups in Agile Software Development: A Review. In: 49th Hawaii International Conference on System Sciences (HICSS), 2016, p. 529-538.
- NEVES, J. L. Pesquisa qualitativa: características, usos e possibilidades. *Caderno de pesquisas em administração*, São Paulo, v. 1, n. 3, p. 1-5, 1996.
- OLIVÉ, A. *Conceptual modeling: foundations and applications*. New York: Springer, 2007.
- PAI, G. et al. Empirical Study of Requirements Engineering Practices in Agile Projects. *Information and Software Technology*, v. 55, n. 11, p. 1997-2012, 2013.
- PATTON, M. *Qualitative research and evaluation methods*. Londres, Thousand Oaks: Sage Publications, 2002.
- RAMOS, Rafael Anderson de Lima. Um Framework Para Adaptação de Processos de Software Guiado por TDD em Aderência a Norma ISO/IEC e IEEE 12207. 2015. 96 f. Monografia (Especialização) - Curso de Bacharelado em Ciência da Computação, Centro Universitário de João Pessoa - Unipê, João Pessoa, 2015.
- RIBEIRO, M.; WERNER, C. M. L.; CHAN, A. (2016). "Developing User Stories with Requirements Engineering Techniques: An Exploratory Study." In: 24th IEEE International Requirements Engineering Conference Workshops (REW). Beijing, China, p. 1-6.

RIBEIRO, M.; WERNER, C. M. L.; CHAN, A. P. C. User Story Map: An Agile User Experience Practice. In: INTERNATIONAL CONFERENCE ON AGILE SOFTWARE DEVELOPMENT. Springer, 2016. p. 110-124.

RICCA, Alessandro. Mockups as a Tool for Software Requirements Elicitation: A Systematic Mapping Study. In: IEEE 12th International Conference on Industrial Informatics (INDIN), 2014, p. 1046-1051.

RIVERO, Alejandro M. Evaluation of Design Models in the Requirements Elicitation Phase. In: 22nd IEEE International Requirements Engineering Conference (RE), 2014.

RIVERO, J.M., GRIGERA, J., ROSSI, G., LUNA, E. R., MONTERO, F., GAEDKE, M. Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering. Information and Software Technology, Volume 56, Issue 6, June Pages 670-687, ISSN 0950-5849, <http://dx.doi.org/10.1016/j.infsof.2014.01.011>, 2014.

RUNESON, P., MARTIN, H. Guidelines for conducting and reporting case study research in software engineering. Empirical Software. 14, 2, 131-164. <http://dx.doi.org/10.1007/s10664-008-9102-8>, 2009.

SAITO, S., TAKEUCHI, M., HIRAOKA, M., KITANI, T., AOYAMA, M. Requirements clinic: Third party inspection methodology and practice for improving the quality of software requirements specifications. Requirements Engineering Conference (RE), 21st IEEE International, pp.290-295. doi:10.1109/RE.2013.6636732, 2013.

SCHWABER, K. Agile Project Management with Scrum. Redmond, WA: Microsoft Press, 2001.

SCHWABER, Ken. The Enterprise and Scrum. 1st. ed. Washington: Microsoft Press, 2007.

SCHWABER, Ken; SUTHERLAND, Jeff. Scrum-opas. Creative Commons, 2017.

SOMMERVILLE, Ian. Engenharia de Software. 9ª ed. Pearson, 2011.

SOMMERVILLE, Ian. Engenharia de Software. 10o ed.: Pearson, São Paulo, 2019.

TERENCE, A. C. F.; ESCRIVÃO FILHO, E. Abordagem quantitativa, qualitativa e a utilização da pesquisa-ação nos estudos organizacionais. Anais, 2006.

THIOLLENT, M. Metodologia da pesquisa-ação. 15. ed. São Paulo: Cortez, 2007.

TRIPP, D. Pesquisa-ação: uma introdução metodológica. Educação e pesquisa, v. 31, n. 3, p. 443-466, 2005.

VERSIONE, 10th annual State of Agile Report, Version One, 2023. Disponível em: <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>. Acesso em: 08 maio 2023.

VERSIONONE, State of Agile Report, 2021. Disponível em: <https://www.versionone.com/state-of-agile-report/state-of-agile-report-2021/>. Acesso em: 08 maio 2023.

VERSIONE, The 17th Annual State of Agile Report, Technical Report, Version One, 2023. Disponível em: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>. Acesso em: 23 de abril.

WANDERLEY, F., SILVA, A., ARAUJO, J., SILVEIRA, D. S. SnapMind: A framework to support consistency and validation of model-based requirements in agile development. IEEE 4th MoDRE, Karlskrona, Sweden. 2014. WHICHARD, G. Definition of Done vs. Acceptance Criteria. Disponível em: <http://www.governmentciomagazine.com/2014/08/definition-done-vs-acceptance-criteria>. Acesso em: 04 julho 2023.

WANDERLEY, Fernando; SILVA, Antonio; ARAÚJO, João. Evaluation of BehaviorMap: a User-Centered Behavior Language. In: IEEE Xplore. [S.l.], 2015.

WESTBROOK, R. Action research: a new paradigm for research in production and operations management. International Journal of Operations & Production Management, v. 15, n. 12, p. 6-20, 1995.

APÊNDICES

APÊNDICE A – ROTEIRO DA ENTREVISTA PARA COLETA DE DADOS ANTES DA INTRODUÇÃO A ABORDAGEM RSD

Informação do respondente

Projeto	
Nome	
Função	

Introdução

O objetivo desta entrevista é realizar uma pesquisa sobre o estado atual do projeto, a fim de identificar os elementos cruciais que devem ser levados em conta antes da introdução da abordagem RSD e que serão utilizados como parâmetros na avaliação da aplicação da abordagem.

Ficariamos muito gratos por sua contribuição para esta pesquisa. Isso incluiria a participação nesta entrevista.

Todas as suas respostas serão mantidas estritamente confidenciais. A sua participação é voluntária. Você não será, de forma alguma, penalizado se optar por não participar do estudo.

Sobre as respostas

Não há respostas certas ou erradas para a maioria das perguntas desta entrevista. Portanto, responda às perguntas da forma mais espontânea e honesta possível, sabendo que suas respostas não serão, de forma alguma, divulgadas a outras pessoas dentro ou fora de sua empresa.

Questões

Perfil do membro da equipe	
ID	Questões
Q1	Qual a sua função no projeto?
Q2	Qual é a sua formação acadêmica?
Q3	Quantos anos de experiência você tem nesta função?

Agora falaremos sobre o conceito de qualidade da ERS em projetos ágeis.

Conceito de qualidade de ERS	
ID	Questões
Q4	<p>O que você considera uma Especificação de Requisitos de Software (ERS) com qualidade? Ou seja, para codificar um recurso, quais características você gostaria de encontrar na ERS?</p> <p>Sonda: Por quê?</p>
Q5	<p>No projeto atual, como você avalia o impacto dos seguintes fatores na qualidade da ERS e no seu desempenho? Compromete (prejudica) ou não afeta?</p> <ul style="list-style-type: none">• Pouca experiência da equipe em especificação de requisitos• Pouca experiência da equipe com desenvolvimento ágil• Estrutura de documentação utilizada pelo projeto• Uma ERS mais focado no cliente do que na equipe de desenvolvimento• Mudanças regras de negócio <p>Sonda: Por quê?</p>
Q6	<p>Você acha que um ERS dividido em vários artefatos facilita o entendimento ou é mais produtivo para o programador obter informações de forma consolidada?</p>

	Sonda: Por quê?
--	-----------------

Agora falaremos sobre o conteúdo usado para especificar requisitos em seu projeto atual.

Conteúdo da ERS	
ID	Questões
Q7	<p>No projeto atual, quais artefatos são usados por um programador para codificar um recurso?</p> <p>Sonda: Casos de Uso \ Cenários, Critérios de aceitação, Funcionalidades, Histórias de Usuário, Objetivos, Protótipo de Interface, Diagrama de Classes, Diagrama de Atividades, Modelo de Dados, Regras, Sem ERS, ...?</p>
Q8	<p>Que outras informações não estão presentes na ERS usada atualmente, mas são necessárias para codificar um requisito?</p> <p>Sonda: Por quê?</p>
Q9	<p>Você conhece ou já trabalhou com alguma abordagem de especificação de requisitos que seja mais voltada para a equipe de desenvolvimento ao invés de ser voltada para o cliente?</p>
Q10	<p>A ERS é suficiente para o desenvolvedor produzir código, ou o desenvolvedor frequentemente precisa consultar outras fontes de informação (por exemplo, pessoas, documentos, etc.) para responder a perguntas?</p> <p>Sonda: Se a ERS não for suficiente, o que você acha que normalmente falta ao desenvolvedor?</p>
Q11	<p>Você considera que a ERS possui informações desnecessárias que não são úteis para a execução de suas atividades?</p> <p>Sondagem: Se sim, que tipo de informação você considera desnecessária?</p>

As próximas perguntas são sobre como a ERS é descrita em seu projeto atual.

Qualidade de Especificação	
ID	Questões
Q12	<p>No projeto atual, você considera que a ERS é ambígua, ou seja, pouco clara e sujeita a mais de uma interpretação?</p> <p>Sonda: Se SIM, o que você acha que contribui para que a ERS seja escrita dessa forma?</p>
Q13	<p>No projeto atual, você considera que a ERS apresenta problemas de consistência entre os requisitos ou entre os artefatos?</p> <p>Sonda: Se SIM, por que você acha que isso acontece?</p>

Agora falaremos sobre o processo de software usado em seu projeto atual.

Contexto do Processo de Desenvolvimento	
ID	Questões
Q14	Qual é a frequência de comunicação com o cliente para adquirir informações?
Q15	<p>Considera que a comunicação com o cliente é adequada ou compromete o desempenho das suas atividades?</p> <p>Sonda: Por quê?</p>
Q16	<p>O projeto contém informações sobre as relações de dependência entre os requisitos?</p> <p>Sonda: Se sim, como essas informações são atualizadas ao longo do desenvolvimento?</p>

Agora vamos falar sobre suas sugestões para melhorar a ERS.

Sugestões de Melhorias	
ID	Questões
Q17	Que boas práticas poderiam ser utilizadas nas atividades de requisitos em outros projetos que melhorariam o desempenho de suas atividades? Sonda: Por quê? Como?

Questões aplicadas a gerentes de projetos

Oportunidades de melhoria	
ID	Questões
Q1	Atualmente, quantos funcionários trabalham no projeto? Quantos deles trabalham exclusivamente com desenvolvimento de software (análise, implementação, teste, qualidade, gerenciamento de projetos, etc.)?
Q2	Quais práticas ágeis você considera que o projeto utiliza em seu processo de desenvolvimento de software? Há quanto tempo essas práticas são utilizadas no projeto?
Q3	Quais documentos (artefatos) a equipe usa para desenvolver software? Estão disponíveis para os clientes?
Q4	A validação de requisitos é feita por meio de entregas frequentes de software ou por meio de documentação? As validações são realizadas presencialmente com o cliente ou remotamente?
Q5	Em geral, quanto tempo dura cada sprint (ciclo de desenvolvimento)? As validações de software com cliente são feitas ao final de cada sprint? Sonda: Se não, com que frequência?
Q6	Como é feita a análise de impacto quando são solicitadas mudanças nos requisitos de software?

Q7	Na sua opinião, quais são os fatores que afetam a qualidade da ERS? Sonda: Por quê?
Q8	Quais boas práticas poderiam ser utilizadas em projetos para melhorar o desempenho da equipe? Sonda: Como?

APÊNDICE B – ROTEIRO DA ENTREVISTA PARA AVALIAÇÃO DA ABORDAGEM RSD

Informação do respondente

Projeto	
Nome	
Função	

Introdução

O objetivo desta entrevista é avaliar a utilização da abordagem RSD (*Requirements Specification for Developers*) no projeto, quais as melhores práticas, dificuldades e limitações da abordagem proposta. Acreditamos que este estudo é essencial para melhorar a adoção da abordagem na prática.

Ficariamos muito gratos por sua contribuição para esta pesquisa. Isso incluiria a participação nesta entrevista.

Todas as suas respostas serão mantidas estritamente confidenciais. A sua participação é voluntária. Você não será, de forma alguma, penalizado se optar por não participar do estudo.

Sobre as respostas

Não há respostas certas ou erradas para a maioria das perguntas desta entrevista. Portanto, responda às perguntas da forma mais espontânea e honesta possível, sabendo que suas respostas não serão, de forma alguma, divulgadas a outras pessoas dentro ou fora de sua empresa.

Questões

Histórico do membro da equipe	
ID	Questões
Q1	Que papéis você desempenhou no projeto?
Q2	Há quanto tempo você trabalha nesta função?
Q3	Como descreve a sua experiência nesta função?

Agora vamos falar sobre o processo de aprendizagem da abordagem RSD.

Aprendizagem da Abordagem	
ID	Questões
Q4	Como foi o processo de aprendizagem da abordagem RSD, o que ajudou ou dificultou nesse processo?
Q5	O que te ajudou ou dificultou nesse processo de abordagem de aprendizagem? Sonda: Por quê?

As próximas questões são sobre a estrutura e o conteúdo da RSD.

Estrutura e conteúdo do RSD	
ID	Questões
Q6	Como você avalia o RSD estruturado por meio de protótipos, conceitos e critérios de aceitação (o tipo de informação)? Sonda: Por quê?

Q7	<p>Numa escala de um (inadequado) a cinco (muito adequado), como avalia a estrutura do RSD (o tipo de informação)?</p> <p>Sonda: Por quê?</p>
Q8	<p>Usando uma escala de um (inadequado) a cinco (muito adequado), como você avalia a conformidade do RSD com cada fator de qualidade?</p> <p>Simplicidade</p> <p>Informações Consolidadas</p> <p>Informações técnicas</p> <p>Critérios de Aceitação</p> <p>Histórico de alterações</p> <p>Sonda: Por quê?</p>
Q9	<p>A qualidade do RSD foi diferente do que você esperava?</p> <p>Sonda: O quê?</p>

Agora vamos falar sobre o esforço necessário para usar a abordagem RSD.

Esforço necessário para usar RSD	
ID	Questões
Q10	Em relação ao esforço necessário, quais são as diferenças entre o RSD e outras abordagens que você usou antes?
Q11	<p>Quais práticas utilizadas no projeto atrapalham seu desempenho?</p> <p>Sonda: Como?</p>

Q12	Quais práticas utilizadas no projeto melhoram seu desempenho? Sonda: Como?
Q13	Usando uma escala discreta (muito, razoável, pouco), como você avalia o esforço necessário para usar a abordagem RSD? Sonda: Por quê?
Q14	Usando uma escala discreta (inferior, igual, superior), como você avalia o esforço necessário para especificar usando a abordagem RSD em comparação com outras abordagens que você usou antes? Sonda: Por quê?
Q15	Usando uma escala discreta (inferior, igual, superior), como você avalia o esforço necessário para codificar a partir do RSD em comparação com outras abordagens que você usou antes? Sonda: Por quê?
Q16	Usando uma escala discreta (inferior, igual, superior), como você avalia o esforço necessário para testar o RSD em comparação com outras abordagens que você usou antes? Sonda: Por quê?
Q17	O esforço necessário para usar o RSD é diferente do que você esperava? Sonda: Por quê?

Agora vamos falar sobre as mudanças nos requisitos.

Mudanças nos requisitos	
ID	Questões
Q18	Usando uma escala discreta (ruim, normal, boa), como você avalia a estratégia

	<p>adotada para analisar as solicitações de mudança nos requisitos?</p> <p>Sonda: Por quê?</p>
Q19	Como as mudanças afetam seu trabalho?
Q20	<p>Usando uma escala discreta (inferior, igual, superior), como você avalia o impacto dessas mudanças em comparação com outras abordagens que você usou anteriormente?</p> <p>Sonda: Por quê?</p>

Por fim, falaremos sobre oportunidades para melhorar a adoção da abordagem RSD.

Oportunidades de melhoria	
ID	Questões
Q21	<p>Que mudanças você gostaria de fazer na abordagem RSD?</p> <p>Sonda: Por quê?</p>
Q22	<p>Quais são as práticas que você não recomenda usar em projetos futuros?</p> <p>Sonda: Por quê?</p>
Q23	<p>Quais práticas você recomenda usar em projetos futuros?</p> <p>Sonda: Por quê?</p>

ANEXOS

ANEXO C – TERMO DE CONSENTIMENTO DA PESQUISA

Título da Pesquisa: Avaliação da utilização da abordagem RSD para especificar requisitos em uma empresa de software

Pesquisador: Thiago Augusto Firmino Barbosa

Prezado(a):

Sou aluno do Mestrado em Tecnologia da Informação do Instituto Federal da Paraíba e atualmente estou conduzindo uma pesquisa sob a orientação da Professora Doutora Juliana Dantas Ribeiro Viana de Medeiros e coorientação da Professora Doutora Heremita Brasileiro Lira. O objetivo da pesquisa é avaliar a abordagem RSD (*Requirements Specification for Developers*) que propõe uma Especificação de Requisitos de Software para desenvolvedores.

A sua contribuição e dos membros da equipe serão cruciais para a coleta de dados que guiarão este estudo, relacionando aspectos de suas experiências profissionais e pessoais com os temas de Gerenciamento de Projetos e Especificação de Requisitos. Para atingir esse objetivo, utilizaremos vários métodos de coleta de dados, incluindo entrevistas técnicas semiestruturadas, observação direta e pesquisa de registros históricos e documentações do projeto.

A participação neste estudo é completamente opcional e voluntária, portanto, se em algum momento qualquer membro da equipe decidir não participar ou desejar desistir da pesquisa, tem total liberdade para fazê-lo.

Todas as informações coletadas nesta pesquisa são altamente confidenciais. Apenas o pesquisador, a orientadora e a co-orientadora terão acesso a esses dados. Na divulgação dos resultados, sua identidade e dos demais, assim como os dados do projeto, serão mantidos em sigilo absoluto, e todas as informações que possam identificá-los serão omitidas.

Embora não haja benefícios diretos em participar desta pesquisa, você e toda equipe estarão indiretamente contribuindo para o avanço do conhecimento científico e para uma melhor compreensão do fenômeno estudado, tanto no meio acadêmico quanto empresarial.

A participação nesta pesquisa não implicará em nenhum tipo de despesa ou pagamento.

Quaisquer dúvidas relativas à pesquisa poderão ser esclarecidas:

Pesquisador	
E-mail	thiago.firmino@academico.ifpb.edu.br

Telefone	+55 83 98749-5461
----------	-------------------

Orientadora	
E-mail	juliana.medeiros@ifpb.edu.br
Telefone	+55 83 98897-1168

Coorientadora	
E-mail	heremita@ifpb.edu.br
Telefone	+55 83 99982-4889

Após a apresentação dessas informações, pedimos sua autorização para dar seguimento a pesquisa.

Consentimento Livre e Esclarecido

Considerando as informações fornecidas acima, eu concordo voluntariamente em participar da pesquisa de forma esclarecida e consciente, além de autorizar a participação voluntária dos membros da equipe. Confirmo que recebi uma cópia deste termo de consentimento e autorizo a realização da pesquisa, bem como a divulgação das estimativas de dados coletados neste estudo.

Assinatura do Gerente do Projeto

Assinatura do Pesquisador

Assinatura do Orientador

Assinatura do Coorientador