



**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**

SANMARA ALVES ARAÚJO

**PROCESSO PARA ESPECIFICAÇÃO DE
REQUISITOS ATRAVÉS DA
INTEGRAÇÃO DE ATDD E PBB**

DISSERTAÇÃO DE MESTRADO

**JOÃO PESSOA – PB
2026**

Processo Para Especificação de Requisitos Através da Integração de ATDD e PBB

Dissertação de Mestrado apresentada como requisito final 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: Profa. Dra. Juliana Dantas Ribeiro
Viana de Medeiros

Coorientadora: Profa. Dra. Nadja da Nóbrega
Rodrigues

**JOÃO PESSOA – PB
2026**

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

A663p Araújo, Sanmara Alves.

Processo para especificação de requisitos através da integração de ATDD e PBB / Sanmara Alves Araújo. – 2025.

143 f. : il.

Dissertação (Mestrado em Tecnologia da Informação) – Instituto Federal de Educação da Paraíba / Programa de Pós-Graduação em Tecnologia da Informação (PPGTI), 2025.

Orientação: Prof^a Dra. Juliana Dantas Ribeiro V. de Medeiros.

Coorientação: Prof^a. Dra. Nadja da Nóbrega Rodrigues.

1. Especificação de requisitos. 2. ATDD. 3. PBB. 4. Desenvolvimento ágil de *software*. I. Título.

CDU 004.41(043)

SANMARA ALVES ARAÚJO

**PROCESSO PARA ESPECIFICAÇÃO DE REQUISITOS ATRAVÉS DA
INTEGRAÇÃO DE ATDD E PBB**

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 26 de fevereiro de 2026.

Membros da Banca Examinadora:

Profa. Dra. Juliana Dantas Ribeiro Viana de Medeiros

Instituto Federal da Paraíba (IFPB)

Orientadora

Profa. Dra. Nadja da Nóbrega Rodrigues

Instituto Federal da Paraíba (IFPB)

Coorientadora

Prof. Dr. Jose Thiago Holanda de Alcantara Cabral

Universidade Federal da Paraíba (UFPB)

Examinador

Prof. Dr. Katyusco de Farias Santos

Instituto Federal da Paraíba (IFPB)

Examinador

Documento assinado eletronicamente por:

- **Juliana Dantas Ribeiro Viana de Medeiros**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 02/03/2026 12:23:30.
- **Nadja da Nobrega Rodrigues**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 02/03/2026 17:38:31.
- **Sanmara Alves Araújo**, DISCENTE (202219840009) DE Mestrado Profissional em Tecnologia da Informação, em 02/03/2026 17:56:54.
- **Jose Thiago Holanda de Alcantara Cabral**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/03/2026 17:29:02.
- **Katysco de Farias Santos**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 06/03/2026 16:50:39.

Este documento foi emitido pelo SUAP em 02/03/2026. 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 842715
Verificador: 2b19e556a9
Código de Autenticação:



“Tenho em mim todos os sonhos do mundo”.

(Fernando Pessoa)

AGRADECIMENTOS

Agradeço primeiramente a Deus, a minha mãe Sônia e a minha irmã Sâmia, por todo apoio que me foi dedicado, não só no Mestrado, mas em todos os momentos da minha vida.

Agradeço também à minha orientadora, Prof.^a Dra. Juliana Dantas Ribeiro e a minha coorientadora, Prof.^a Dra. Nadja da Nóbrega Rodrigues, pela confiança, incentivo e apoio durante todo o longo percurso deste projeto de pesquisa.

Agradeço a Paula, por muitas vezes me ajudar com palavras de apoio, para seguir firme no meu propósito, de ser mestra.

Agradeço aos meus colegas de trabalho que puderam contribuir para que essa pesquisa se realizasse.

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

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

RESUMO

Apesar da adoção de práticas ágeis ter crescido nos últimos anos como estratégia de desenvolvimento de *software*, a engenharia de requisitos no desenvolvimento de *software* ágil ainda enfrenta alguns desafios, dentre eles: documentação inadequada e com baixo nível de detalhes; requisitos funcionais e não funcionais despercebidos, o que pode aumentar o esforço de manutenção de *software* e elevar os custos dos projetos. Nesse contexto, o principal objetivo desta pesquisa foi investigar a adoção integrada das práticas ATDD (*Acceptance Test-Driven Development*) e PBB (*Product Backlog Building*) como estratégia para execução das atividades de requisitos em projetos ágeis. Para isso, foi realizada uma pesquisa-ação qualitativa em uma empresa de *software*, tendo como principal resultado a definição e implantação de um processo para especificar requisitos em projetos ágeis através da integração sistemática das práticas ATDD e PBB. Entrevistas semiestruturadas, observações diretas no processo de desenvolvimento e análise de documentos foram os instrumentos utilizados para coleta dos dados. O processo definido foi implantado na empresa sendo monitorado durante três ciclos de desenvolvimento. A utilização do processo na empresa demonstrou melhoria na organização do *backlog*, e na clareza e detalhamento dos critérios de aceitação. Além de favorecer a comunicação entre os membros da equipe. Entretanto, algumas dificuldades iniciais foram identificadas, como a curva de aprendizado e adaptação com a metodologia *Scrum*. Conclui-se que o processo proposto apresenta potencial de aplicabilidade prática, contribuindo para a melhoria das atividades de engenharia de requisitos em ambientes ágeis.

Palavras-chaves: Especificação de Requisitos; ATDD; PBB; Desenvolvimento Ágil de *Software*.

ABSTRACT

Although the adoption of agile practices has grown in recent years as a software development strategy, requirements engineering in agile software development still faces some challenges, including: inadequate documentation with a low level of detail; overlooked functional and non-functional requirements, which can increase software maintenance effort and raise project costs. In this context, the main objective of this research was to investigate the integrated adoption of ATDD (Acceptance Test-Driven Development) and PBB (Product Backlog Building) practices as a strategy for executing requirements activities in agile projects. To this end, qualitative action research was conducted in a software company, with the main result being the definition and implementation of a process for specifying requirements in agile projects through the systematic integration of ATDD and PBB practices. Semi-structured interviews, direct observations of the development process, and document analysis were the instruments used for data collection. The defined process was implemented in the company and monitored during three development cycles. The use of the process in the company demonstrated improvement in backlog organization and in the clarity and detail of acceptance criteria. In addition to facilitating communication among team members, some initial difficulties were identified, such as the learning curve and adaptation to the Scrum methodology. It is concluded that the proposed process shows potential for practical applicability, contributing to the improvement of requirements engineering activities in agile environments.

Keywords: *Requirements Specification; ATDD; PBB; Agile Software Development.*

LISTA DE FIGURAS

Figura 1 - Composição das etapas da pesquisa-ação	20
Figura 2 - Custo relativo para correção de erros e defeitos	24
Figura 3 - The Agile Subway Map	27
Figura 4 - PBB Canvas	28
Figura 5 - PBB Canvas preenchido	29
Figura 6 - Ciclo de desenvolvimento ATDD	31
Figura 7 - Processo de síntese temática	45
Figura 8 - História de Usuário no PBI	47
Figura 9 - Casos de Testes	48
Figura 10 - Protótipos no Figma	48
Figura 11 - Visão Geral do IAP	52
Figura 12 – Atividades do Subprocesso “Construção do Backlog do Produto”	54
Figura 13 - Bloco “Product Name” no PBB Canvas	55
Figura 14 - Bloco “Problems” no PBB Canvas	56
Figura 15 - Bloco do “Expectations” no PBB Canvas	56
Figura 16 - Exemplo Preenchimento dos blocos Product Name, Problems e Expectations no PBB Canvas	57
Figura 17 - Exemplo de Preenchimento do bloco Persona no PBB Canvas	58
Figura 18 - Fluxo de Preenchimento da seção Features	58
Figura 19 - Exemplo do Fluxo de Preenchimento da seção Features	59
Figura 20 - Exemplo de representação textual no modelo ARO	60
Figura 21 - Fluxo de trabalho passo a passo	61
Figura 22 - Evolução com perguntas, comentários e ideias	62
Figura 23 - Itens de um Backlog do Produto	62
Figura 24 - Técnica de Priorização - COORG	64
Figura 25 - Priorização do Backlog	65
Figura 26 - Classificar o Backlog	66
Figura 27 - Ordem de Prioridade	66
Figura 28 - Ordenar o Backlog	67
Figura 29 - Organizar o Backlog	67
Figura 30 - Preparação do Backlog da Sprint	70
Figura 31– Refinamento das Histórias de Usuário em BDD	73
Figura 32 - Sequência básica do TDD	78
Figura 33 - Desenvolvimento do Sistema	79
Figura 34 - Exibir o histórico de pedidos corretamente	80
Figura 35 - Filtrar pedidos por status	80
Figura 36 - Histórico de pedidos sem pedidos realizados	81
Figura 37 - Desenvolvimento dos Testes Automatizados	84
Figura 38 - Estrutura de Pastas do Robot	85
Figura 39 - Exemplo da estrutura inicial do arquivo.robot	86
Figura 40 - Exemplo de implementação	87
Figura 41 - Revisão das Funcionalidades Criadas	90
Figura 42 - Exemplo do log do framework Robot	91
Figura 43 – Campos utilizados para o Apêndice B no Azure Devops	96

Figura 44 – Campo utilizado para o Apêndice C no Azure Devops	97
Figura 45 - Percentuais avaliados no critério de completude.....	105
Figura 46 – Conformidade do Processo IAP por Fator de Qualidade	106
Figura 47 - Percentuais avaliados no critério de legibilidade.....	107
Figura 48 - Percentuais avaliados no critério de objetividade	108
Figura 49 - Percentuais avaliados no critério de volatilidade	108

LISTA DE TABELAS

Tabela 1- Definições de Requisitos	22
Tabela 2 - Tarefas da engenharia de requisitos	24
Tabela 3 – Etapa de Discussão	32
Tabela 4 - Etapa de Detalhar	33
Tabela 5 – Etapa de Desenvolver	34
Tabela 6 – Etapa de Demonstrar	35
Tabela 7 - Caracterização da Empresa Estudada	39
Tabela 8 - Caracterização do Projeto Analisado	40
Tabela 9 - Caracterização dos Participantes da Pesquisa	41
Tabela 10 - Pontos Positivos Identificados nas Entrevistas	49
Tabela 11 - Frequência dos Problemas Identificados na Fase Inicial	50
Tabela 12 - Visão Geral da Construção do Backlog do Produto	63
Tabela 13 - Visão Geral da Priorização do Backlog	68
Tabela 14 - Visão Geral da Preparação do Backlog da Sprint	72
Tabela 15 - Visão geral do Refinamento das Histórias de Usuário para o BDD	76
Tabela 16 - Visão geral do Desenvolvimento do Sistema	83
Tabela 17 - Resumo do Desenvolvimento dos Testes Automatizados	89
Tabela 18 - Resumo da Revisão das Funcionalidades Criadas	92

LISTA DE ABREVIATURAS E SIGLAS

ATDD	<i>Acceptance Test-Driven Development</i>
BDD	<i>Behavior Driven Development</i>
COORG	<i>Classificar, Ordenar e ORGANizar</i>
DOD	<i>Definition of done</i>
DOR	<i>Definition of ready</i>
ER	<i>Engenharia de Requisitos</i>
IBM	<i>International Business Machines</i>
PBB	<i>Product Backlog Building</i>
PBI	<i>Product Backlog Item</i>
PO	<i>Product Owner</i>
QA	<i>Quality Assurance</i>
TDD	<i>Test Driven Development</i>

SUMÁRIO

1. INTRODUÇÃO	17
1.1. Motivação e Definição do Problema	17
1.2. Objetivos	18
<i>1.2.1. Objetivo geral</i>	18
<i>1.2.2. Objetivos específicos</i>	18
1.3. Metodologia	19
1.4. Aplicabilidade	20
1.5. Estrutura do Documento	20
2. REFERENCIAL TEÓRICO	22
2.1. Requisitos de Software	22
2.2. Desenvolvimento Ágil de Software	25
<i>2.2.1 Product Backlog Building (PBB)</i>	27
<i>2.2.2 Acceptance Test-Driven Development (ATDD)</i>	30
2.3. Trabalhos Relacionados	36
3. INVESTIGAÇÃO SITUAÇÃO INICIAL DA EMPRESA	38
3.1. Iniciação da Pesquisa-Ação	38
<i>3.1.1 Definição do Tema e Interessados</i>	41
<i>3.1.2. Delimitação do Problema</i>	42
<i>3.1.3. Definição dos Critérios de Avaliação da Pesquisa-Ação</i>	43
3.2. Procedimentos para Coleta, Análise e Síntese dos Dados	43
3.3. Resultados da Avaliação	46
<i>3.4.1. Avaliação do Processo de Engenharia de Requisitos</i>	46
<i>3.4.2. Pontos Positivos Identificados</i>	49
<i>3.4.3. Principais Problemas Identificados</i>	50
<i>3.4.4. Oportunidades de Melhoria Identificadas</i>	51
4. PROCESSO PARA INTEGRAÇÃO DAS PRÁTICAS ATDD e PBB	52
4.1. Visão geral do Processo para Integração de ATDD e PBB (IAP)	52
4.2. Detalhamento do Processo	53
<i>4.2.1. Construção do Backlog do Produto</i>	54
4.2.1.1 Contextualizar o Produto	55
4.2.1.2. Descrever as Personas	57

4.2.1.3. Entender as Funcionalidades	58
4.2.1.4 Identificar os PBIs	59
4.2.2. <i>Priorização do Backlog</i>	64
4.2.2.1. Classificar o Backlog	65
4.2.2.2. Ordenar o Backlog	67
4.2.2.3. Organizar o Backlog	67
4.2.3. <i>Preparação do Backlog da Sprint</i>	68
4.2.3.1. Criar a História de Usuário	70
4.2.3.2. Definir os Critérios de Aceite	70
4.2.3.3. Verificar Necessidade de Protótipo	71
4.2.3.4. Descrever Protótipo	71
4.2.3.5. Identificar as Dependências	71
4.2.3.6. Refinar com o Time	72
4.2.4 <i>Refinamento das Histórias de Usuário para o BDD</i>	73
4.2.4.1. Revisar a História do Usuário	74
4.2.4.2. Criar os Cenários em BDD	74
4.2.4.3. Inserir os Cenários na História do Usuário	75
4.2.4.4. Preparar para Automação e Testes	76
4.2.5. <i>Desenvolvimento do Sistema</i>	77
4.2.5.1. Criar os Testes Unitários	79
4.2.5.2. Codificar a Funcionalidade	81
4.2.5.3. Validar os Testes	82
4.2.5.4. Verificar Correção do Código	82
4.2.5.5. Corrigir o Código	82
4.2.6. <i>Desenvolvimento dos Testes Automatizados</i>	83
4.2.6.1. Criar os Testes de Aceitação	84
4.2.6.2. Validar os Testes	87
4.2.6.3. Verificar se os Critérios de Aceitação foram atendidos	88
4.2.6.4. Corrigir o Teste ou a Funcionalidade	88
4.2.6.5. Refatorar o Teste ou a Funcionalidade	89
4.2.7 <i>Revisão das Funcionalidades Criadas</i>	90
4.2.7.1. Demonstrar a Funcionalidade	90
4.2.7.2. Exibir os Critérios de Aceitação Atendidos	91
4.2.7.3. Verificar Reajuste nos Critérios de Aceitação ou Funcionalidade	92

4.2.7.4. Voltar para Subprocesso Preparação do <i>Backlog</i> da <i>Sprint</i>	92
4.2.7.5. Incrementar no Produto	92
5. IMPLANTAÇÃO DO PROCESSO IAP	94
5.1. Contexto da implantação	94
5.2. Capacitação no Processo	94
5.2.1. <i>Estratégia adotada para a Capacitação</i>	95
5.2.2. <i>Dificuldades e Lições Aprendidas Durante a Capacitação</i>	97
5.3. Monitoramento da Utilização do Processo	98
5.4. Dificuldades Identificadas na Implantação	100
6. AVALIAÇÃO DO PROCESSO IAP	103
6.1. Coleta e Análise dos dados	103
6.1.1. <i>Conteúdo</i>	104
6.1.2. <i>Estrutura</i>	109
6.2. Comparação do Processo IAP em Relação às Práticas Anteriores	110
6.2.1. <i>Esforço Necessário para Compreender, Codificar e Testar</i>	110
6.2.2. <i>Esforço Necessário para Estruturação</i>	111
6.2.3. <i>Análise de Impacto de Pedidos de Mudança</i>	112
6.2.4. <i>Transferência de Conhecimento, Colaboração do Cliente e Dependência entre Stakeholders</i>	113
6.3. Resultados da Avaliação dos Critérios de Pesquisa-Ação	114
6.4. Síntese Temática dos Resultados	116
6.5. Reflexões sobre os resultados	118
7. CONCLUSÃO	120
7.1. Contribuições da Pesquisa	120
7.2. Trabalhos Futuros	121
7.3. Limitações da pesquisa e Ameaças à Validade	122
APÊNDICES	127
APÊNDICE A – ROTEIRO DA ENTREVISTA PARA COLETA DE DADOS ANTES DA INTRODUÇÃO AO PROCESSO IAP	128
APÊNDICE B – TEMPLATE PARA CRIAÇÃO DE HISTÓRIA DE USUÁRIO	135
APÊNDICE C – TEMPLATE PARA CRIAÇÃO DO BDD – CENÁRIOS DE TESTE (QA)	137
APÊNDICE D – ROTEIRO DA ENTREVISTA PARA AVALIAÇÃO DO PROCESSO IAP	139
ANEXOS	145
ANEXO A – TERMO DE CONSENTIMENTO DA PESQUISA	146

1. INTRODUÇÃO

Atualmente, empresas de desenvolvimento de *software* estão cada vez mais focadas em acelerar a entrega de projetos sem comprometer a qualidade. Pesquisas indicam que um dos principais fatores que levam ao fracasso de projetos de *software* está relacionado a erros na fase de Engenharia de Requisitos, pois esses erros, se não corrigidos, afetam negativamente as etapas seguintes (FAGUNDES et al., 2020).

Segundo Sales (2022), pesquisas recentes mostram que quanto mais tarde um erro é identificado no sistema, mais caro e complexo é corrigi-lo. Por isso, é essencial que cada fase do desenvolvimento de *software* seja bem planejada e executada. Se os requisitos não forem claramente definidos e implementados corretamente, tanto as organizações desenvolvedoras quanto os clientes podem sofrer prejuízos, afetando tanto a qualidade quanto os custos do projeto.

Conforme Longo (2020), estima-se que 85% dos defeitos de *softwares* desenvolvidos tenham origem em requisitos ambíguos, incompletos e ilusórios. Devido às ambiguidades e complexidades da linguagem natural, técnicas textuais, e do processo de captura, os requisitos apresentam muitas vezes inconsistências, redundância, incompletude e omissões.

No contexto ágil, também há desafios constantes relacionados à especificação e testes de requisitos, como por exemplo, a comunicação inadequada com os testadores em relação às modificações de requisitos pode levar a uma verificação incorreta de requisitos desatualizados ou à falha na verificação de novos requisitos, o que poderia comprometer a qualidade do software. As atividades da Engenharia de Requisitos em projetos ágeis ainda são marcadas pela variedade limitada de práticas ágeis que apoiem plenamente os processos de elicitação, análise, especificação, validação e gerenciamento de requisitos (MIRANDA, 2024).

1.1. Motivação e Definição do Problema

Embora a adoção de métodos ágeis tenha atenuado alguns problemas no desenvolvimento de *software*, como a falta de flexibilidade e a comunicação ineficaz entre as partes interessadas em modelos tradicionais, os desenvolvedores ainda enfrentam diversas dificuldades, como por exemplo, a escassez de informações sobre requisitos e a documentação insuficiente dos mesmos, já que os métodos ágeis priorizam a entrega de software funcional em detrimento da documentação abrangente, o que pode levar a um entendimento inadequado dos requisitos em projetos de médio a grande porte.

As mudanças contínuas nos requisitos também são um desafio constante, exigindo que as equipes se adaptem rapidamente. Outro ponto crítico é a dificuldade em definir o escopo do projeto no início, o que, combinado à priorização contínua dos requisitos, pode levar a um desvio de escopo se não for bem gerenciado (INAYAT et al, 2015).

Segundo Medeiros (2015), Histórias de Usuários é a técnica mais comumente utilizada para especificação de requisitos em projetos ágeis. As histórias de usuário são descrições curtas e simples de uma funcionalidade do sistema sob a perspectiva do usuário final, geralmente estruturadas no formato “Como [tipo de usuário], eu quero [ação] para [benefício]”. Elas têm como objetivo representar necessidades do usuário de forma compreensível para toda a equipe, servindo como base para o desenvolvimento e validação das funcionalidades.

De acordo com Aguiar e Caroli (2021), a utilização do *Product Backlog Building* (PBB) no desenvolvimento ágil de *software* oferece benefícios, como por exemplo: permite desenvolver um entendimento compartilhado da necessidade do produto; facilita a descoberta do produto e a concepção das histórias de usuários; estrutura a granularidade dos itens do *backlog*; possibilita priorizar os itens de maior importância e valor para o cliente; e melhora o trabalho do time com a criação e o refinamento do *backlog*.

Segundo Rebelo (2025), as vantagens do *Acceptance Test-Driven Development* (ATDD) no desenvolvimento ágil incluem: favorecer uma maior colaboração e comunicação entre todos os envolvidos no desenvolvimento de um produto; proporcionar um entendimento mais claro e refinado dos requisitos; e possibilitar um acordo sobre o que será desenvolvido durante uma iteração/sprint entre os principais *stakeholders*, alinhando o resultado às expectativas do cliente.

Apesar da escassez de estudos empíricos que avaliem as práticas ATDD e PBB, elas demonstram ter potencial para serem utilizadas como uma estratégia para alinhar o desenvolvimento de *software* aos objetivos de negócio de forma colaborativa e eficiente nos levantamentos de requisitos.

1.2. Objetivos

1.2.1. Objetivo geral

O objetivo geral desta pesquisa é investigar, de forma sistemática, a adoção integrada das práticas ATDD (*Acceptance Test-Driven Development*) e PBB (*Product Backlog Building*) como estratégia para execução das atividades de requisitos em projetos ágeis.

1.2.2. Objetivos específicos

Os objetivos específicos do presente trabalho são:

- Investigar como as atividades de requisitos estão sendo realizadas em uma empresa de *software*, buscando identificar suas características, práticas utilizadas, bem como os principais desafios no desenvolvimento ágil de *software*;

- Definir um processo, detalhando os subprocessos e as atividades necessárias para adoção sistemática e integrada das práticas ATDD e PBB como estratégia para especificar requisitos de *software* em projetos ágeis;
- Implantar o processo proposto na empresa;
- Avaliar a adoção do processo proposto para identificar o impacto nas atividades de especificação de requisitos, detalhando os pontos fortes e as limitações;

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

- Como as práticas ATDD e PBB podem ser utilizadas de forma integrada e sistemática como estratégia para especificar requisitos em projetos ágeis, atendendo às equipes de desenvolvimento e qualidade?

1.3. Metodologia

A investigação sobre as práticas ATDD e PBB foi realizada através de uma pesquisa-ação qualitativa, com o propósito exploratório e de natureza aplicada. Esta estratégia foi escolhida considerando a necessidade de a pesquisadora interferir no processo atual de desenvolvimento de software de uma empresa.

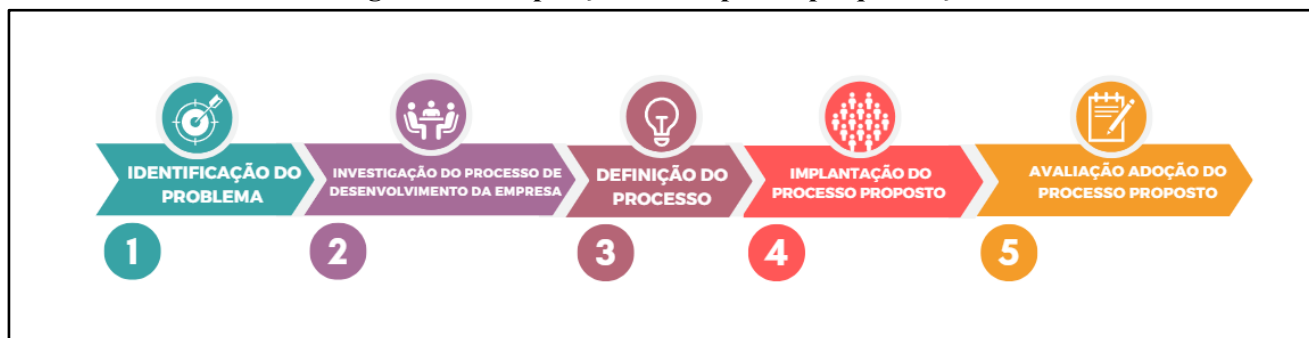
A pesquisa-ação confere, aos dados obtidos e observados, sempre um caráter descritivo e rico em significados, considerando contexto/ambiente natural em que se desenvolve a investigação (CORRÊA, DE CAMPOS, e ALMAGRO, 2018).

Segundo Thiollent (2005), o método é definido como sendo uma pesquisa social baseada no modo empírico, identificada e executada através de uma ação ou com a resolução de um problema, no qual os pesquisadores e participantes representantes da situação ou do problema, estejam envolvidos de modo cooperativo e participativo.

Mediante uma abordagem de pesquisa, o método Pesquisa-Ação é aplicado para gerar novos conhecimentos através da pesquisa de solução ou melhorias para situações problemáticas da realidade (MCKAY e MARSHALL, 2001).

As etapas para execução deste estudo foram adaptadas de Mckay e Marshall (2001), e estão ilustradas na Figura 1.

Figura 1 - Composição das etapas da pesquisa-ação



Fonte: Modificado do trabalho de McKay e Marshall (2001).

- Identificação do Problema: Etapa inicial que teve por objetivo a identificação do problema que justificasse a realização da pesquisa, conforme descrito na seção 1.1.
- Investigação do processo de desenvolvimento da empresa: Avaliar como as atividades de requisitos estavam sendo realizadas em uma empresa de software com o propósito de identificar as boas práticas, desafios e limitações no processo adotado na empresa.
- Definição do Processo: Descrever como as práticas ATDD e PBB podem ser utilizadas como estratégia para solução do problema. Como parte da definição do processo são descritas as atividades, sequência de utilização, entradas, saídas e ferramentas necessárias.
- Implantação do Processo Proposto: Capacitar a equipe de desenvolvimento sobre como utilizar o processo proposto.
- Avaliação adoção do Processo Proposto: Acompanhar a adoção do processo proposto durante três *sprints* para coletar dados sobre como os *stakeholders* avaliam a utilização, os pontos fortes e as limitações.

1.4. Aplicabilidade

O processo proposto nesta pesquisa tem o potencial de ser utilizado por empresas de *software* que utilizam práticas ágeis no seu processo de desenvolvimento, podendo ser adaptado para atender às particularidades de cada empresa.

1.5. Estrutura do Documento

O documento está organizado da seguinte forma:

- O capítulo 1 trata da introdução, contextualizando a temática em que a pesquisa está inserida e os objetivos que pretende alcançar, bem como a metodologia da pesquisa-ação, escolhida por possibilitar uma investigação participativa e iterativa no ambiente estudado.
- O capítulo 2 apresenta a fundamentação teórica dos conteúdos abordados na pesquisa, além de uma breve análise dos trabalhos relacionados com esta pesquisa.

- O capítulo 3 aborda a avaliação que foi realizada na empresa como estratégia para fornecer subsídios à definição do processo para integração do ATDD e PBB.
- O capítulo 4 apresenta o processo proposto para integração do ATDD e PBB.
- O capítulo 5 descreve a implantação do processo proposto.
- O capítulo 6 detalha uma a avaliação do processo após ter sido implantado na empresa.
- Por fim, o capítulo 7 relata a conclusão do trabalho com contribuições da pesquisa e trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta os principais conceitos que fundamentaram a realização desta pesquisa. Inicialmente, conceitua-se requisito, sua classificação, sua importância no processo de desenvolvimento de *software* e o impacto das falhas nos requisitos. Em seguida, apresenta-se os processos e principais características da Engenharia de Requisitos (ER), mostrando os benefícios da sua aplicação em projetos de *software*. Em seguida, é apresentada a fundamentação sobre as metodologias ágeis de desenvolvimento, e sobre as práticas ATDD e PBB.

2.1. Requisitos de Software

Segundo Wiegers e Beatty (2013), requisitos são especificações das funções que devem ser implementadas, descrições de como o software deve se comportar, ou uma propriedade ou atributo do mesmo. Os requisitos de software podem ser classificados quanto ao tipo como funcional e não funcional (BOURQUE; FAIRLEY et al., 2014).

Os requisitos funcionais são declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações, como também podem explicitamente declarar o que o sistema não deve fazer. Requisitos não funcionais são restrições sobre os serviços ou as funções oferecidas pelo sistema, restrições de tempo, padrões, segurança, entre outros (SOMMERVILLE, 2018).

Para Figueira (2012, p.16): “um dos principais objetivos da engenharia de requisitos é encontrar requisitos adequados para o sistema em desenvolvimento”, e questiona o real significado do requisito. Para responder tal questionamento, Figueira (2012), traz uma série de definições na Tabela 1.

Tabela 1- Definições de Requisitos

Autor	Definição
Tayer e Dorfmann (2000)	Requisito é uma característica do software necessária para que o usuário e/ou cliente possa encontrar a solução de um problema de forma a atingir um objetivo.
Sommerville (2007)	Requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais. Os requisitos demonstram as necessidades de um cliente de um sistema que ajuda a resolver um determinado problema.
Lopes apud Siddiqi (1996)	Requisitos são como uma declaração completa do que o software irá fazer sem referir-se a como fazê-lo.

Lopes apud Kruchten (2000)	Um requisito é como uma condição ou capacidade que um software deve realizar.
Goguem (1997)	Requisitos são propriedades que um software deve ter para funcionar com êxito no ambiente em que for utilizado.

Fonte: Figueira (2012, p.16).

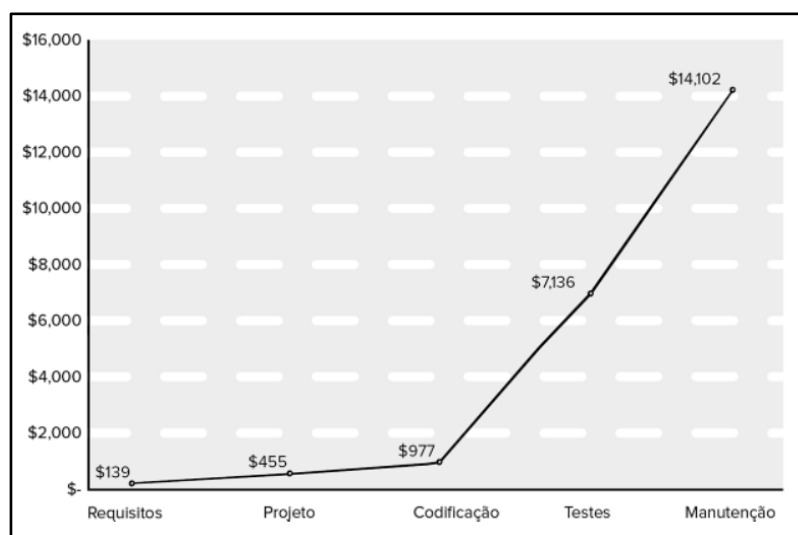
Nesse contexto, percebe-se que os requisitos expressam as características necessárias que o software deve ter para a realização das funcionalidades, as quais satisfaçam os objetivos dos *stakeholders*, ou seja, pessoas envolvidas no projeto ou que são afetadas pelo sistema em desenvolvimento como clientes, usuários, gerentes de projeto e dentre outros).

Problemas de requisitos em um projeto gera um produto com má qualidade, e quanto antes forem detectadas eventuais falhas, o custo de sua correção será menor. Assim, uma falha encontrada na fase de requisitos tem um custo menor do que uma identificada na fase de manutenção. Um exemplo do que tal situação representa monetariamente, é descrito por Pressman e Maxim (2021, p. 318), em seu estudo sobre a empresa Cigital:

De acordo com os dados médios do setor, o custo para descobrir e corrigir defeitos durante a fase de codificação é de US\$977 por defeito. Portanto, o custo total para corrigir os 200 defeitos “críticos” durante essa fase (200 x US\$977) é de aproximadamente US\$195.400. Os dados médios do setor mostram que o custo para descobrir e corrigir defeitos durante a fase de testes é de US\$7.136 por defeito. Nesse caso, supondo que a fase de testes do sistema tenha revelado aproximadamente 50 defeitos críticos (ou apenas 25% daqueles encontrados pela Cigital na fase de codificação), o custo para descobrir e corrigir esses defeitos (50 x US\$ 7.136) teria sido de aproximadamente US\$356.800.

O exemplo de Pressman e Maxim (2021) é corroborado por Sommerville (2018) ao mencionar que o maior problema enfrentado durante o desenvolvimento de um software é a ER. A evolução do custo de correção de um software é progressiva diante do avanço das fases em que vai se completando, conforme exposto na Figura 2, a seguir:

Figura 2 - Custo relativo para correção de erros e defeitos



Fonte: Pressman e Maxim (2021).

O desenvolvimento de *software* com qualidade necessita de compreensão dos problemas, o que ocorre na fase de requisitos. Assim, infere-se o número de problemas por meio da adoção de um processo formal de ER, o qual permita uma real compreensão das necessidades para o sistema, com a delimitação correta dos processos a serem implementados e a validação das necessidades do cliente (PRESSMAN e MAXIM, 2021).

Conforme Pressman e Maxim (2021), a engenharia de requisitos abrange sete tarefas distintas, são elas: a concepção, levantamento, elaboração, negociação, especificação, validação e gestão. Tendo como objetivo fornecer aos *stakeholders* um entendimento por escrito do projeto ou problema, como mostrado na Tabela 2 (PRESSMAN e MAXIM, 2021).

Tabela 2 - Tarefas da engenharia de requisitos

Etapas	Definições
Concepção	Estabelece um entendimento básico do problema, das pessoas que querem uma solução e da natureza da solução desejada.
Levantamento	Define os requisitos básicos do projeto.
Elaboração	Expande ainda mais os requisitos básicos em um modelo baseado em cenários, classes e atividades.
Negociação	Alinha as reais necessidades para que o objetivo do projeto seja obtido e definição de prioridades.
Especificação	Documento por escrito, cenários de uso, um protótipo ou qualquer combinação dos fatores citados.

Validação	Todos os requisitos precisam ser validados em relação às necessidades do cliente para garantir que o sistema correto será construído.
Gerenciamento de Requisitos	Conjunto de atividades que ajuda a equipe de projeto a identificar, controlar e acompanhar as necessidades de suas mudanças à medida que o projeto prossegue.

Fonte: Adaptado a partir de Pressman e Maxim (2021, pg. 104).

O propósito da engenharia de requisitos é fazer a ponte entre o solicitador do sistema e a equipe de desenvolvimento. Com isto, a ER mostra-se como uma das partes mais importantes do processo de desenvolvimento, uma vez que os requisitos mal elaborados podem gerar *bugs* e/ou o não atendimento das necessidades do cliente.

2.2. Desenvolvimento Ágil de Software

O desenvolvimento ágil de software é norteado pelo manifesto ágil (BECK, 2000) que define os valores que devem ser considerados:

- Indivíduos e interações mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

A partir destes valores foram definidos os 12 princípios por trás do manifesto ágil (BECK, 2000):

1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
3. Entregar frequentemente software funcionando, em intervalos de poucas semanas a poucos meses, com preferência à menor escala de tempo.
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto durante todo o projeto.
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
6. Informações para e entre uma equipe de desenvolvimento é através de conversa face a face.
7. Software funcionando é a medida primária de progresso.

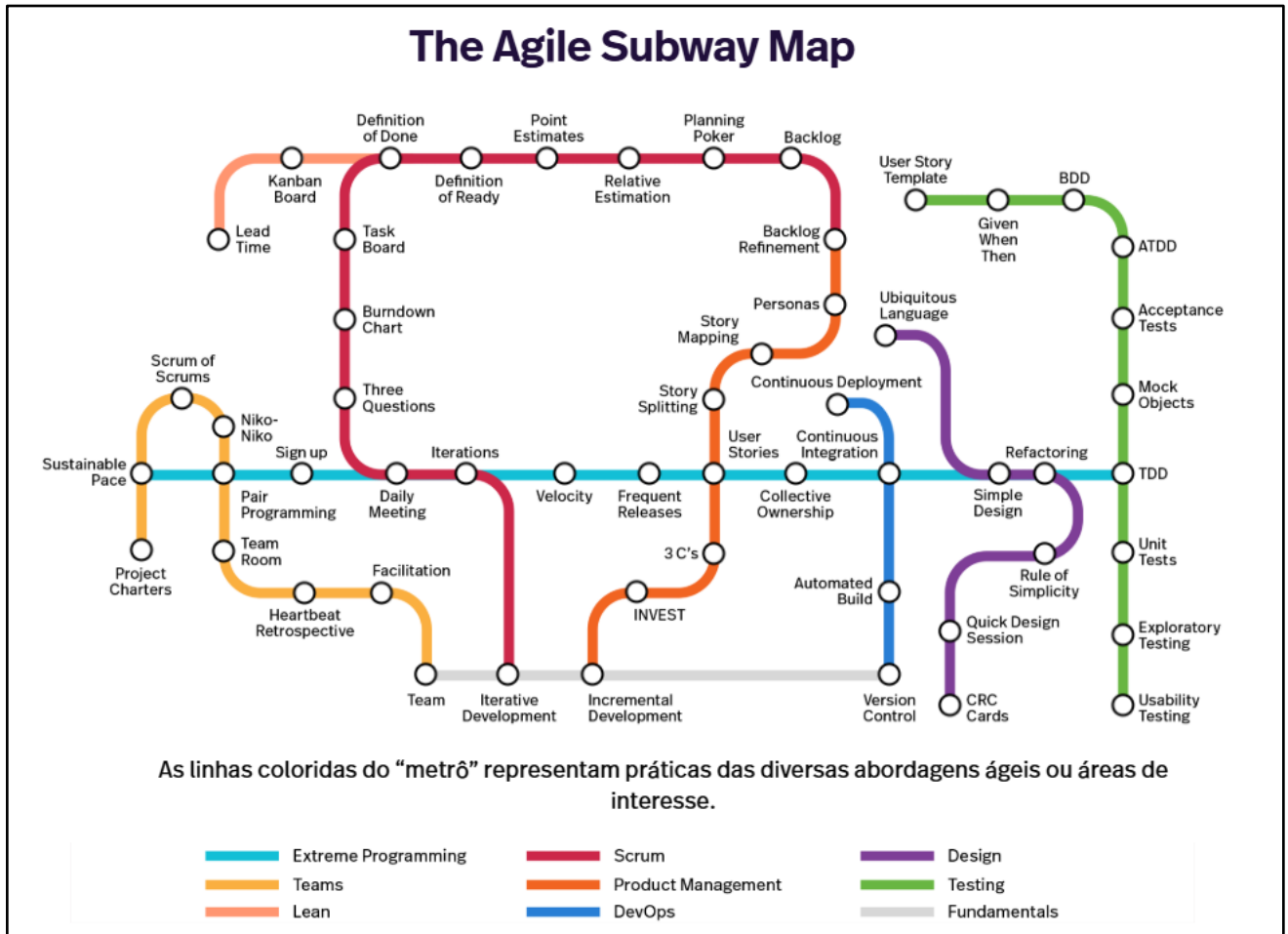
8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9. A contínua atenção à excelência técnica e bom *design* aumenta a agilidade.
10. A simplicidade, a arte de maximizar a quantidade de trabalho não realizado, é essencial.
11. As melhores arquiteturas, requisitos e *designs* emergem de equipes auto-organizáveis.
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Uma das metodologias ágeis mais conhecidas é o *Scrum*, que organiza o trabalho em iterações curtas e limitadas no tempo, conhecidas como *sprints*, nas quais um conjunto de atividades é planejado, executado e avaliado. O *Scrum* também define papéis bem estabelecidos, como o *Scrum Master*, responsável por facilitar o processo, remover impedimentos e garantir a correta aplicação das práticas ágeis, e o *Product Owner* (PO), responsável por gerenciar o *backlog* do produto, definir prioridades e assegurar que as entregas estejam alinhadas às necessidades do negócio e dos *stakeholders*. Essas estruturas são cruciais para a gestão de projetos complexos, pois facilitam a adaptabilidade e o progresso iterativo, alinhando-se à satisfação do cliente e à entrega contínua de valor (DARAOJIMBA et al., 2024).

Em complemento, as Práticas Ágeis são as técnicas operacionais e estratégias essenciais empregadas diariamente para concretizar a metodologia escolhida e garantir a qualidade do produto final. A Programação Extrema (XP), por exemplo, enfatiza práticas de engenharia como o Desenvolvimento Orientado a Testes (TDD) e a Programação em Pares para melhorar a excelência técnica. Além disso, a implementação ágil bem-sucedida depende do envolvimento contínuo das partes interessadas e do *feedback* frequente para garantir que o produto atenda às necessidades em evolução do cliente. No geral, essas práticas promovem uma cultura de colaboração, transparência e melhoria contínua dentro da equipe, fatores que aprimoram a agilidade geral da organização (DARAOJIMBA et al., 2024).

A vasta coleção dessas práticas e como elas se interconectam é ilustrada pelo "*The Agile Subway Map*", uma metáfora visual onde as linhas representam grandes áreas de conhecimento (*Extreme Programming* ou *Scrum*) e as estações são as práticas específicas (como ATDD e Backlog), guiando as equipes em sua jornada de adoção ágil e na busca pela melhoria contínua, conforme exposto na Figura 3 (AGILE ALLIANCE, 2025).

Figura 3 - The Agile Subway Map



Fonte: Agile Alliance (2025).

2.2.1 Product Backlog Building (PBB)

O *Product Backlog Building* (PBB) – tem como finalidade ajudar no refinamento e construção do *Product Backlog* de forma colaborativa. Sua dinâmica consiste em vivenciar na prática, a elaboração e criação de um *backlog* colaborativo e efetivo, juntando todas as pessoas que trabalharão no produto e utilizando o PBB Canvas como ferramenta de facilitação (AGUIAR e CAROLI, 2021).

Criada por Paulo Aguiar, em 2021, a dinâmica do PBB consiste em um processo de construção do *backlog* utilizando o PBB Canvas como ferramenta de facilitação. Embora tenha sido originado em times que utilizavam o *framework* SCRUM a fim de resolver o *Product Backlog*, trata-se de uma ferramenta que também pode ser complementar para times que utilizam outros *frameworks* e metodologias ágeis, sendo considerada base para o desenvolvimento da facilitação por meio de um fluxo linear de trabalho. Através da proposta de elaboração e criação de *backlog* efetivo e colaborativo, a ferramenta reúne pessoas que trabalham direta ou indiretamente no produto (AGUIAR e CAROLI, 2021).

A dinâmica leva todos os envolvidos no negócio a uma prática de elaboração e definição do *backlog* efetivo, consistente e alinhado com os valores de negócio do cliente. O PBB Canvas tem um fluxo bem simples e de fácil compreensão, na Figura 4 é possível verificar um painel, onde serão colocadas todas as informações necessárias para a construção dos requisitos, e logo abaixo, o que cada bloco do painel PBB Canvas representa (AGUIAR e CAROLI, 2021).

Figura 4 - PBB Canvas

PRODUCT BACKLOG BUILDING PBB Canvas		PRODUCT NAME
PROBLEMS	PERSONAS	
	FEATURES	
EXPECTATIONS	PBI: PRODUCT BACKLOG ITEMS	

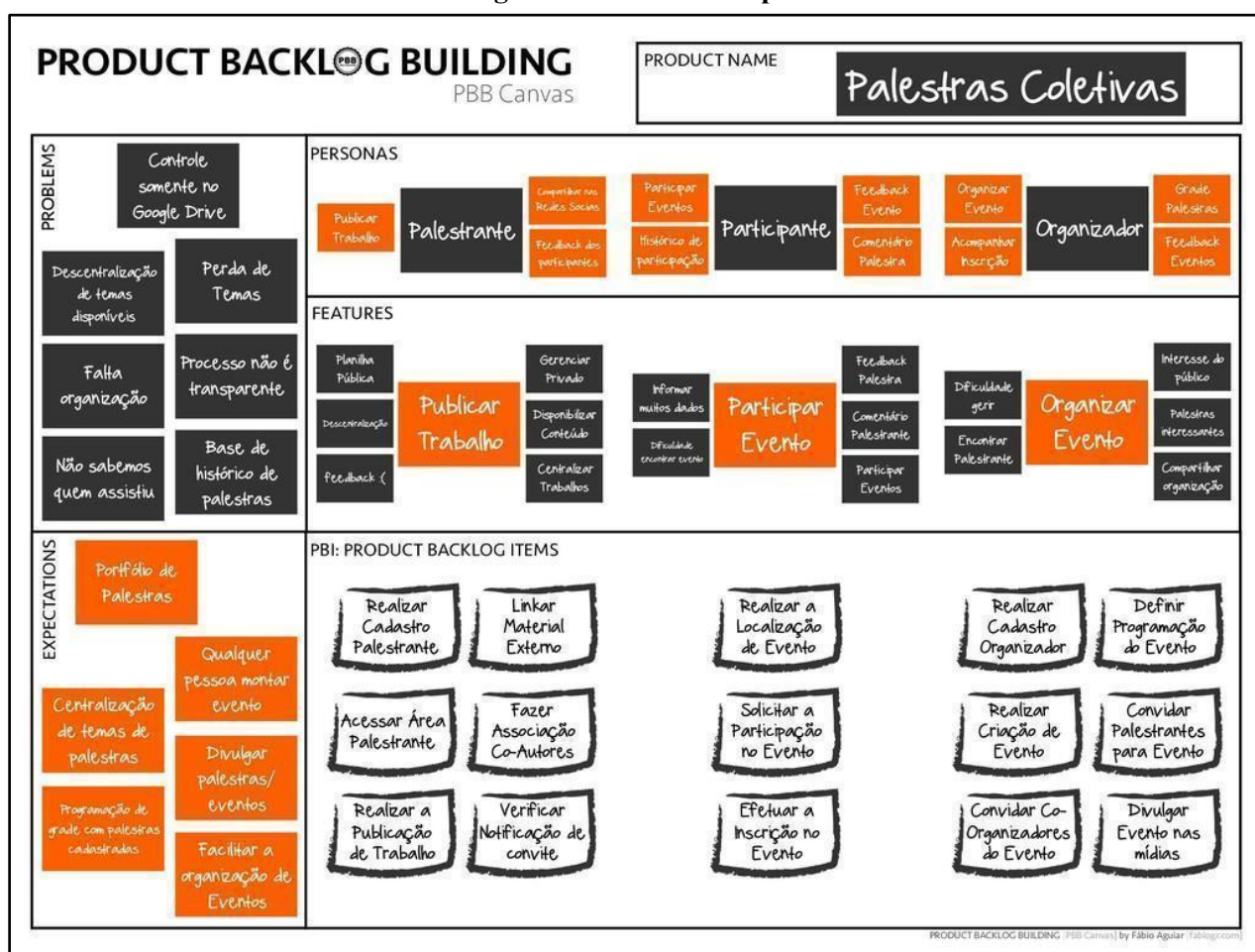
Fonte: Aguiar e Caroli (2021).

- **Product Name:** o primeiro passo é identificar o produto que será construído.
- **Problems:** o segundo passo é compreender os problemas que desejam solucionar com a construção do produto. Neste momento, as partes interessadas buscam, de forma colaborativa, a compreensão dos problemas a serem solucionados e que desejam que sejam resolvidos. É importante conhecer e entender os problemas antes da criação da solução.
- **Expectations:** nesta etapa, deve-se compreender o desejo das partes interessadas com a criação da solução, sempre alinhando suas expectativas aos problemas encontrados no passo anterior.
- **Personas:** identificar os usuários e suas atividades de acordo com seu perfil, o que faz e o que espera poder fazer ao utilizar o produto em construção.
- **Features:** nesta etapa, para cada *persona* elencada no passo anterior, identificar as ações e/ou as interações no produto. Através de perguntas como “Quais ações ou iterações o usuário terá no produto? Quais problemas da persona essa funcionalidade resolve? Quais benefícios essa funcionalidade traz para essa persona?” Através dessas perguntas, é possível obter uma lista de funcionalidades e seus PBIs (*Product Backlog Item*).

- **PBI:** são elementos que compõem o *Product Backlog*. Eles refletem o trabalho de desenvolvimento para melhorar a visão do produto, atendendo as necessidades do cliente ou das partes interessadas. Para cada funcionalidade, deve-se escrever os respectivos PBIs atendendo o passo a passo da funcionalidade até a sua completude.

As vantagens de utilizar o método PBB na construção de um fluxo de organização geral do negócio é que pelo fluxo ser muito simples, ajuda a organizar a visão geral, alinhando assim o valor do negócio, compreende o que o produto irá agregar ao ser entregue e tem a facilidade de ter a ferramenta PBB Canvas disponível, onde apresenta uma visão da construção do *backlog* do produto de forma visualmente organizada, como mostrado na Figura 5 (AGUIAR e CAROLI, 2021).

Figura 5 - PBB Canvas preenchido



Fonte: Aguiar e Caroli (2021).

É possível verificar na Figura 5, que o PBB canvas está dividido em seis blocos, cada bloco preenchido com exemplos de informações correspondente ao seu bloco específico.

2.2.2 *Acceptance Test-Driven Development (ATDD)*

A prática *Acceptance Test-Driven Development (ATDD)* se propõe a promover a colaboração entre cliente, desenvolvedor e testador na criação de novos testes de aceitação. Essa estratégia visa auxiliar na definição de novos requisitos antes de sua implementação (HOFFMAN et al., 2014).

Segundo Cruz (2018), o ATDD se concentra em testes sob a ótica do cliente, com o objetivo de atender às necessidades de negócio. Moe (2019), reforça que a prática ATDD é uma abordagem colaborativa em que usuários, testadores e desenvolvedores definem critérios de aceitação automatizados logo no início do processo de desenvolvimento. Essa prática potencializa que todos os membros do projeto compreendam com clareza o que deve ser feito e implementado.

A prática ATDD visa criar testes automatizados antes mesmo de criar o código que representa as expectativas de comportamento que o software deveria ter. Ao invés de criar testes focados em código com a perspectiva do desenvolvedor, que é o foco do *Test Driven Development (TDD)*, a prática ATDD prega que sejam desenvolvidos testes de aceitação com uma visão direta de negócio voltado ao ponto de vista do usuário.

Para utilização da prática, toda a equipe se reúne em uma plataforma comum para discutir detalhadamente os critérios de aceitação e é então destilado em testes de aceitação concretos. Isso acontece antes de começar a fase de desenvolvimento. A prática ATDD se propõe a colaborar para que todos os *stakeholders* tenham o mesmo entendimento compartilhado dos requisitos. Também permite que a equipe de desenvolvimento entre no projeto com uma imagem clara das necessidades do usuário final, em vez de na interpretação dos indivíduos no meio (SWAMINATHAN, 2024).

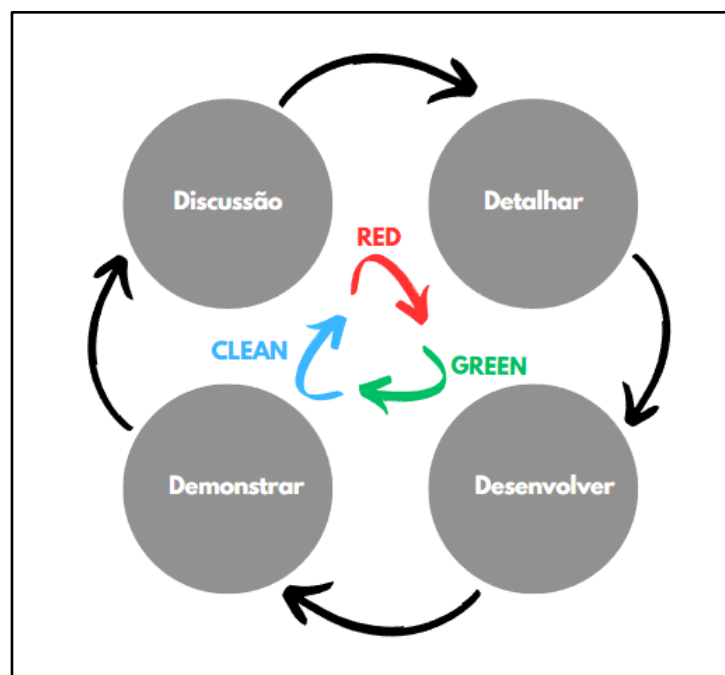
Para entender melhor sobre a prática ATDD é necessário posicioná-la dentre alguns dos tipos de testes existentes. Durante o desenvolvimento do software, os tipos mais comumente empregados (ROCHA, 2023) são de unidade, sistema, integração e de aceitação, os quais são resumidamente explicados abaixo:

- teste de unidade: é realizado durante a codificação, e de forma independente, tem a finalidade de testar cada entidade do sistema. Realiza-se testes em pequenas unidades do software, como métodos, funções e procedimentos. Eles são especialmente importantes para garantir a integridade do software após alterações e refatorações de código, sendo muito utilizados em TDD. Os testes unitários e de aceitação são complementares, não sendo recomendado que um deles seja suprimido após a adoção do outro.
- teste de integração: permite realizar testes combinando diferentes componentes para garantir que continuem funcionando de forma integrada. É um nível de teste que complementa o teste de unidade quando realizado pelo desenvolvedor. Caso seja realizado por um testador, poderá ser necessário utilizar alguma ferramenta para automatizar os testes.

- teste de sistema: o objetivo do teste é identificar falhas relacionadas às regras de negócio especificadas para o sistema ou falhas de execução do sistema. Este teste não necessita ter acesso ao código-fonte para ser realizado. Por isso, os testadores podem realizar os testes.
- testes de aceitação: Os testes de aceitação estão diretamente associados às histórias de usuário, sendo utilizados para validar se os critérios de aceitação definidos foram atendidos. Neste trabalho, os testes de aceitação são a base da abordagem proposta e são detalhados ao longo do texto. De forma resumida, eles validam os critérios de aceitação de uma história de usuário e exigem a participação ativa dos *stakeholders*, tanto na discussão quanto na definição dos testes. O principal desafio no desenvolvimento desses testes é garantir que os requisitos estejam totalmente claros no início da iteração, evitando erros decorrentes de falhas no entendimento. Os testes de aceitação são a base deste trabalho e são detalhados ao longo dele. Resumidamente pode-se dizer que eles validam o critério de aceitação de uma história de usuário. Eles exigem participação ativa do usuário, tanto na sua discussão quanto na escrita de testes. O desafio principal no desenvolvimento dos testes de aceitação é garantir que os requisitos estejam totalmente claros no início da iteração, evitando erros no desenvolvimento devido a falhas no entendimento dos requisitos.

O ciclo de ATDD, conforme ilustrado na Figura 6 é orientado a teste de aceitação e é formado por quatro etapas.

Figura 6 - Ciclo de desenvolvimento ATDD



Fonte: Adaptado de Massari (2018).

A fase de discussão ocorre em uma reunião com o objetivo de coletar os critérios de aceitação do cliente. Segundo Cruz (2018), uma forma eficaz de conduzir essa discussão é por meio de uma

reunião de planejamento da interação de negócios com o cliente. Com os critérios de aceitação definidos, a etapa de Detalhamento permite a documentação dos testes de forma padronizada e formal, alinhada às necessidades do cliente. Massari (2018) ressalta que a documentação nesta fase deve ser acessível e incorporada a *frameworks* que facilitem a integração com a automação de testes. Assim, na fase de desenvolvimento, os testes automatizados são escritos com base no fluxo estabelecido anteriormente. Finalmente, na etapa de Demonstração, o cliente deverá validar se os requisitos atendem aos critérios de aceitação previamente definidos nas etapas anteriores.

A seguir são detalhadas as etapas da Figura 6:

- **Discussão:** Nesta etapa se discutem os requisitos e alinham-se as expectativas. As histórias de usuário (*user stories*) são refinadas em um *workshop* ou em uma reunião de preparação do *backlog* do produto (*backlog refinement*), antes da reunião de planejamento da iteração/*sprint*. Em ambos os casos, os participantes são uma equipe multifuncional, o *Product Owner* e, algum outro interessado que potencialmente tem mais informações sobre as histórias.
- **Detalhar:** Nesta etapa os requisitos são refinados e convertidos para a linguagem suportada pelo *framework* de teste.
- **Desenvolver:** Após o refinamento, a aplicação deve ser desenvolvida utilizando TDD com base nos cenários definidos. Em paralelo, os testes de aceitação devem ser automatizados a partir dos critérios de aceitação e dos cenários BDD elaborados anteriormente, permitindo a validação contínua da funcionalidade.
- **Demonstrar:** Na etapa final desta iteração do desenvolvimento da história, o software funcionando é mostrado para os clientes, que opinam sobre as funcionalidades desenvolvidas, podendo levantar alterações para atender às suas expectativas.

A seguir apresentamos um exemplo detalhado que utiliza a funcionalidade de "Cadastro de Novo Usuário" para ilustrar as quatro etapas do ATDD (Discussão, Detalhar, Desenvolver e Demonstrar):

- **Etapa Discutir - Entendimento Compartilhado**

A equipe se reúne (*Product Owner*, Desenvolvedor e Testador/QA) para alinhar o entendimento sobre a *User Story* "Como usuário, quero me cadastrar no sistema para acessar funcionalidades exclusivas.". Abaixo Tabela 3, com o exemplo da etapa de discussão.

Tabela 3 – Etapa de Discussão

Participante	Foco da Discussão
<i>Product Owner</i>	O Quê: O formulário deve exigir e-mail (único), nome completo e uma senha segura.

Desenvolvedor	Como: Quais validações técnicas são necessárias? O e-mail deve ser salvo no banco de dados. Qual a complexidade de criptografia da senha?
Testador (QA)	Cobertura de Testes: O que acontece se o e-mail já existir? O que define uma "senha segura" (mínimo de 8 caracteres, letras e números)? O que acontece se faltar um campo?
Saída	
A equipe concorda que o Critério de Aceitação principal é: "O usuário deve ser capaz de criar uma conta com sucesso fornecendo um e-mail único, nome e uma senha forte, sendo imediatamente logado no sistema."	

Fonte: Próprio Autor.

- **Etapa Detalhar - Formalizando os Exemplos**

O entendimento verbal é convertido em cenários de teste concretos, geralmente usando a sintaxe *Gherkin*¹, que servem como a especificação formal, conforme ilustrado na Tabela 4:

Tabela 4 - Etapa de Detalhar

Cenário de Aceitação (Gherkin)	Propósito
<p>Cenário 1: Cadastro Básico de Sucesso</p> <p>Dado que estou na página de cadastro.</p> <p>Quando preencho Nome, e-mail único e Senha válida.</p> <p>Então minha conta é criada com sucesso e sou logado.</p>	<p>Cenário Positivo: Caminho feliz e esperado.</p>
<p>Cenário 2: E-mail Já Cadastrado (Falha)</p> <p>Dado que existe um usuário com o e-mail "teste@exemplo.com".</p> <p>Quando tento me cadastrar com o e-mail "teste@exemplo.com".</p> <p>Então vejo a mensagem de erro: "E-mail já cadastrado."</p>	<p>Cenário Negativo: Tratamento de validação de unicidade.</p>

¹ <https://cucumber.io/docs/gherkin/>

<p>Cenário 3: Senha Fraca (Falha)</p> <p>Dado que estou na página de cadastro.</p> <p>Quando insiro uma senha com menos de 8 caracteres.</p> <p>Então vejo a mensagem de erro: "A senha deve ter no mínimo 8 caracteres."</p>	<p>Cenário de Borda: Garantir a validação de segurança.</p>
Saída	
<p>Um conjunto de três Testes de Aceitação formais, claros e não ambíguos.</p>	

Fonte: Próprio Autor.

- **Etapa Desenvolver - Código Guiado por Teste Colaborativo**

Nesta fase, o Desenvolvedor e o QA trabalham em paralelo para construir a solução e os testes que a validarão, conforme ilustrado na Tabela 5:

Tabela 5 – Etapa de Desenvolver

Atividade do Desenvolvedor	Atividade do Testador (QA)
<p>TDD (Unidade/Integração): Começa a escrever o código da camada de serviço e do repositório de usuários, sempre começando com testes unitários (<i>RED</i>), escrevendo o mínimo de código para passar (<i>GREEN</i>) e refatorando o código.</p>	<p>Construção do <i>Framework</i> de Teste: O QA pega os cenários <i>Gherkin</i> da Etapa 2 e os mapeia para o código de automação usando uma ferramenta (ex: <i>Robot Framework</i>). Neste ponto, o QA executa os testes e eles falham (<i>RED</i>), pois a funcionalidade ainda não foi implementada.</p>
<p>Prioriza a implementação da regra de negócio (salvar no banco, criptografar a senha, verificar a unicidade do e-mail).</p>	<p>Foca na conexão da automação com a <i>interface</i> do usuário (ou API) e na criação de dados de teste específicos para cada cenário (ex: um usuário para o Cenário 2).</p>
<p>Conclusão: Quando o desenvolvedor acha que a história está pronta, ele executa os Testes de Aceitação do QA. Se todos passarem (<i>GREEN</i>), o desenvolvimento da <i>Story</i> é considerado completo.</p>	<p>Conclusão: O QA acompanha a passagem dos testes, garantindo que a automação seja robusta e o ambiente de teste esteja estável para a Etapa 4.</p>

Saída
O código da funcionalidade está completo e validado por um conjunto de testes de aceitação que passaram na automação.

Fonte: Próprio Autor.

- **Etapa Demonstrar - Validação e *Feedback***

O time demonstra a funcionalidade ao *Product Owner* para validação final de negócio, conforme ilustrado na Tabela 6:

Tabela 6 – Etapa de Demonstrar

Ações Chave
<p>Execução dos Testes: O QA executa a suíte de testes de aceitação automatizados (Cenários 1, 2 e 3) na frente do <i>Product Owner</i> para provar que a funcionalidade atende às regras definidas.</p> <p>Demonstração Manual: O Desenvolvedor ou QA demonstra manualmente o "Cenário Básico de Sucesso" e os cenários de erro diretamente na <i>interface</i> do usuário.</p> <p>Aceitação: O <i>Product Owner</i> verifica que o usuário é de fato logado após o cadastro (Critério 1) e que as mensagens de erro são claras (Critérios 2 e 3). A <i>User Story</i> é aceita e considerada concluída, podendo ser lançada para produção.</p>
Saída
A <i>User Story</i> de "Cadastro de Novo Usuário" é formalmente aceita pelo negócio.

Fonte: Próprio Autor.

O processo de ATDD, embora detalhado em quatro etapas, é iterativo e colaborativo. A aplicação de todos esses passos, desde a discussão inicial com o cliente até a validação final, demonstra como o processo se integra perfeitamente ao *mindset* ágil de entrega de valor e adaptação contínua.

Diversos autores na literatura ágil mencionam o uso do ATDD em combinação com *frameworks* de gerenciamento de processos ágeis, como o Scrum. Tanto Cohn (2011) quanto Cruz (2018) apontam vantagens em integrar o ATDD com o Scrum, uma vez que essa abordagem contribui para o desenvolvimento de funcionalidades já direcionadas ao aceite, com testes pré-definidos pelo cliente. Cruz ainda destaca que o ATDD e o Scrum atuam em sincronia na entrega de valor ao cliente, tornando essas práticas fortes aliadas.

2.3. Trabalhos Relacionados

Apesar de não termos identificados estudos que abordem a utilização das práticas ATDD e PBB de forma integrada, encontramos estudos que avaliam, de maneira isolada, essas e outras práticas que se propõem a atuar nas atividades de requisitos em projetos ágeis. Nesse processo, quatro estudos principais foram identificados e selecionados por apresentarem um alinhamento direto com os objetivos deste trabalho: (DA SILVA et al., 2020), (LISBÔA et al., 2021), (FERREIRA et al., 2021) e (LONGO,2020).

Da Silva et al. (2020) fizeram um estudo de caso relatando a aplicação do método ágil REACT-M (*Requirements Evolution in Agile Context - Management*) para gerência de requisitos em um ambiente real de desenvolvimento de *software*. A pesquisa avaliou as forças, fraquezas, oportunidades e ameaças dos componentes do REACT-M, como artefatos, ciclo de vida, papéis e cerimônias. Os resultados demonstraram que o método é eficaz, simples, colaborativo, adequado, centrado no usuário e orientado a metas, cumprindo seu objetivo de gerenciar a evolução de requisitos de forma iterativa e eficiente no contexto ágil. O estudo oferece uma contribuição prática para a indústria e um entendimento empírico para a comunidade científica sobre a relação entre requisitos e métodos ágeis.

Lisbôa et al. (2021) fizeram um estudo observacional sobre as adaptações ao trabalho remoto no contexto da pandemia de covid-19, os dados foram analisados, e verificou que atividades desenvolvidas por grupo de pessoas permitiram a construção e manutenção do entendimento comum para elaboração dos requisitos. Devido ao cenário da pandemia, estas ferramentas que funcionam em ambientes físicos, também se mostraram fundamentais no ambiente remoto. Entre as ferramentas utilizadas, podemos citar as ferramentas para gestão visual das dinâmicas ágeis com recursos para facilitação de colaboração entre os participantes, com destaque para: Web Mural para as dinâmicas colaborativas remotas de elicitação de requisitos, utilizando o *Lean Inception* e *Product Backlog Building* (PBB).

Por outro lado, em Ferreira et al. (2021) realizaram um estudo de caso que investigou o problema de não envolver os usuários finais nos estágios iniciais de levantamento de requisitos por meio de entrevistas com clientes e usuários finais de um projeto ágil utilizando *Lean Inception* (LI), *Design Thinking* e *Product Backlog Building* (PBB). Os resultados mostram que, embora o LI ajude na identificação de objetivos de negócio e o PBB na construção do *backlog* do produto, há falta de detalhamento sobre as necessidades dos usuários e requisitos de software, impactando negativamente a criação de jornadas do usuário e *personas*. A participação direta dos usuários finais, quando presente, melhora o *Design Thinking* e sua adoção, destacando a importância de envolver os usuários desde o início, em vez de apenas representantes.

De acordo com Longo (2020), o aumento da adoção de práticas ágeis no desenvolvimento de *software* trouxe desafios na especificação e teste de requisitos. O ATDD em conjunto com o *Gherkin* e Cenários de Usuários via Diagramas de Interação do Usuário (US-UIDs), foram adotados para

melhorar a colaboração entre partes interessadas e a qualidade do *software*. Esses testes envolveram "dados funcionais" e "dados de teste", essenciais para a comunicação e automação. A uniformidade dos dados de teste é fundamental para a automação eficiente, mas nem sempre é bem avaliada durante a especificação. Sua pesquisa propôs métricas para medir a uniformidade dos dados de teste, com o objetivo de melhorar a qualidade dos testes de aceitação. As métricas foram implementadas computacionalmente e testadas em projetos controlados, mostrando que o formato *Gherkin* apresenta menos irregularidades.

O trabalho realizado através da pesquisa que vem sendo apresentada se diferencia dos demais pelo objetivo de propor um processo para especificação de requisitos através da integração de dois métodos ágeis: o ATDD e o PBB, adotando uma metodologia de pesquisa-ação, e assim permitindo uma intervenção direta e um acompanhamento contínuo durante a avaliação, fornecendo evidências sobre todo o processo de implementação e construindo assim um entendimento compartilhado da necessidade do produto, facilitando a escrita da história do usuário e do cenário de teste.

É importante destacar que, ao analisar os trabalhos relacionados, esta pesquisa se posiciona para contribuir com as discussões existentes no contexto atual, onde práticas eficazes de especificação de requisitos são cruciais, ao apresentar diferenciais em sua abordagem. 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. INVESTIGAÇÃO SITUAÇÃO INICIAL DA EMPRESA

Este capítulo detalha como foi feita a etapa de iniciação da pesquisa-ação, com foco na investigação do processo de desenvolvimento de *software* em uma empresa do setor de tecnologia da informação. A empresa analisada é caracterizada como de médio porte, atuante no desenvolvimento de soluções de *software* sob demanda, inserida no contexto de desenvolvimento ágil e com atuação no mercado nacional. A escolha da organização se deu pela presença de desafios relacionados às atividades de engenharia de requisitos, especialmente no que se refere à especificação, comunicação e validação de requisitos.

A investigação tem como objetivo avaliar o processo atual de desenvolvimento adotado pela empresa, identificando práticas utilizadas, limitações e oportunidades de melhoria, de modo a fornecer subsídios para a definição de um processo integrado entre ATDD e PBB.

3.1. Iniciação da Pesquisa-Ação

De acordo com Mello et al. (2012), a pesquisa-ação pode ser iniciada de duas formas distintas. A iniciação dirigida pela pesquisa, ocorre quando o pesquisador identifica uma lacuna na literatura acadêmica e busca um objeto de estudo para investigar e preencher essa lacuna. Por outro lado, a iniciação dirigida pelo problema, acontece quando membros de uma empresa enfrentam um problema que não sabem como resolver e, portanto, procuram um especialista teórico para encontrar uma solução.

Neste trabalho, a iniciação do projeto de pesquisa-ação ocorreu de forma dirigida pela pesquisa, começando com uma avaliação detalhada das práticas de requisitos existentes em uma empresa de *software*. O objetivo dessa fase foi estabelecer as bases conceituais e práticas necessárias para a definição de um processo que integre de forma sistemática as práticas ATDD e PBB. Mello et al. (2012) recomenda as seguintes etapas descritas como partes fundamentais do processo de iniciação do projeto de pesquisa-ação:

- **Definição da Estrutura Conceitual-Teórica:** A estrutura conceitual-teórica envolve uma investigação na literatura, buscando estudos que forneçam um panorama da área em estudo. Ao revisar trabalhos clássicos e recentes, é possível identificar áreas que podem se beneficiar de pesquisas mais aprofundadas (MELLO et al., 2012).

Os resultados dessa investigação foram apresentados ao longo dos capítulos 1 e 2. No Capítulo 1, foram também estabelecidos a questão de pesquisa e os objetivos do trabalho. A fundamentação teórica, por sua vez, foi desenvolvida no Capítulo 2, onde foi realizado o

mapeamento da literatura e abordados em detalhes os conceitos relevantes das três áreas principais de pesquisa: Requisitos de Software, ATDD e PBB.

- **Definição da Unidade de Análise:** A pesquisa-ação teve como objeto de estudo, um projeto ágil em andamento em uma empresa de software brasileira, que oferece serviços para automação de softwares voltados para transações no comércio exterior. Com mais de 26 anos de mercado, a empresa está localizada em Campinas, São Paulo e possui funcionários que trabalham remotamente em várias cidades do Brasil.

A Tabela 7 apresenta a caracterização da empresa estudada, com o objetivo de contextualizar o ambiente no qual a pesquisa-ação foi conduzida.

Tabela 7 - Caracterização da Empresa Estudada

Aspecto	Descrição
Tipo de organização	Empresa de desenvolvimento de <i>software</i>
Porte	Médio porte
Área de atuação	Automação de sistemas para comércio exterior
Mercado de atuação	Nacional
Modelo de desenvolvimento	Ágil (Scrum)
Tempo de atuação	Mais de 26 anos
Distribuição da equipe	Remota
Estrutura de equipe	Times multidisciplinares (PO, QA, Desenvolvedores, Gerente de Projetos)
Contexto de requisitos	Uso de <i>backlog</i> de produto e histórias de usuário
Principais desafios	Especificação de requisitos, comunicação e critérios de aceitação

Fonte: Próprio Autor.

O projeto analisado consiste em um sistema em desenvolvimento no contexto ágil, voltado para o domínio de automação de processos relacionados ao comércio exterior. Trata-se de um projeto em andamento, conduzido por meio de iterações (sprints), no qual são aplicadas práticas do framework Scrum. A Tabela 8 apresenta a caracterização do projeto analisado.

Tabela 8 - Caracterização do Projeto Analisado

Aspecto	Descrição
Tipo de projeto	Desenvolvimento de <i>software</i> sob demanda
Domínio de negócio	Comércio exterior
Situação do projeto	Em andamento
Modelo de desenvolvimento	Ágil (Scrum)
Duração das <i>sprints</i>	15 dias
Tamanho da equipe	7 Membros
Papéis envolvidos	<i>Product Owner</i> , QA, Desenvolvedores, Gerente de Projetos
Artefatos utilizados	<i>Backlog</i> do produto, histórias de usuário, critérios de aceitação
Foco da análise	Especificação de requisitos e uso de ATDD e PBB

Fonte: Próprio Autor.

O objetivo foi investigar como os profissionais de uma empresa de tecnologia, em particular um time de desenvolvimento (PO, Analistas de Qualidade, Desenvolvedores e Gerente de Projetos), lidam com especificação de requisitos, e como atuam com as práticas ATDD ou o PBB em projetos de softwares ágeis.

Cabe ressaltar ainda, que embora o projeto e suas atividades de especificação de requisitos façam parte do objeto de estudo, esta pesquisa tem como objetivo principal avaliar o processo de integração do ATDD e PBB através do time de desenvolvimento do projeto. Sendo assim, a unidade de análise é o time de desenvolvimento.

Adicionalmente, a identidade da empresa e dos participantes foi mantida em sigilo, com o objetivo de preservar a confidencialidade das informações.

Os dados analisados nesta pesquisa foram coletados diretamente no ambiente real de desenvolvimento, por meio da observação das atividades do time, participação em cerimônias ágeis e análise de artefatos do projeto, como histórias de usuário, backlog e critérios de aceitação. A utilização da pesquisa-ação permitiu a atuação direta no contexto organizacional, garantindo que as evidências obtidas sejam provenientes de práticas reais, e não de situações hipotéticas.

- **Definição do Contexto e Propósito:** A pesquisa é de natureza exploratória, consistindo em descobrir o campo de pesquisa, os interessados e suas expectativas e em estabelecer um

primeiro diagnóstico da situação, dos problemas prioritários e de eventuais ações (THIOLLENT, 2007).

3.1.1 Definição do Tema e Interessados

Segundo Coughlan e Coughlan (2002), essa etapa é guiada por duas questões relacionadas à racionalidade da ação e da pesquisa. A racionalidade para a ação refere-se ao processo em que a pesquisa-ação ocorre em tempo real, permitindo que os principais membros da organização compreendam o contexto do projeto. Por outro lado, a racionalidade para a pesquisa envolve a análise das razões para estudar essa ação específica, a justificativa para a escolha da pesquisa-ação como o método apropriado e a contribuição esperada para o avanço do conhecimento.

Esta dissertação foca na racionalidade para a ação, uma vez que se desenvolve em tempo real, permitindo que os membros-chave do projeto, tanto da equipe de desenvolvimento quanto dos responsáveis pela empresa, compreendam o sistema em estudo. Dessa forma, todos os envolvidos têm a oportunidade de gerar e assimilar o conhecimento produzido e alcançar a contribuição esperada. Nesta etapa foi essencial definir o grupo de pesquisadores e membros da unidade de análise que colaborarão na condução da pesquisa, na coleta de dados e na implementação de ações para enfrentar o problema identificado (MELLO et al., 2012). A seleção dos participantes foi realizada de forma intencional, considerando critérios relacionados às suas funções no projeto, experiência prática e envolvimento direto com atividades de engenharia de requisitos.

Além disso, buscou-se contemplar profissionais com diferentes níveis de experiência (júnior, pleno e sênior), bem como diferentes formações acadêmicas e tempo de atuação na área, de modo a obter uma visão mais abrangente das práticas adotadas na organização.

A caracterização dos participantes da pesquisa foi realizada contemplando aspectos como papel no projeto, nível de experiência, tempo de atuação, formação acadêmica e envolvimento com atividades de requisitos. A Tabela 9 apresenta essa caracterização de forma detalhada.

Tabela 9 - Caracterização dos Participantes da Pesquisa

Participante	Papel no Projeto	Nível	Experiência (anos)	Formação Acadêmica	Atuação em Requisitos
P1	Desenvolvedor <i>Back-End</i>	Sênior	10+	Sistemas de Informação	Participa do refinamento técnico e implementação das histórias de usuário.
P2	<i>Product Owner</i>	Júnior	2	Administração	Atua na definição e priorização do backlog, valida critérios de aceitação

P3	Desenvolvedor <i>Front-End</i>	Sênior	10+	Tecnólogo de Gestão de TI	Participa do refinamento técnico e implementação das histórias de usuário.
P4	Desenvolvedor <i>Back-End</i>	Sênior	10+	Ciência da Computação	Participa do refinamento técnico e implementação das histórias de usuário
P5	Analista de Requisitos	Júnior	2	Sistemas de Informação	Define as histórias de usuário.
P6	Gerente de Projetos	Sênior	10+	Ciência da Computação	Atua na gestão do projeto e alinhamento estratégico dos requisitos
P7	Analista de Qualidade	Pleno	4	Analista de Qualidade	Executa testes e auxilia na validação dos requisitos

Fonte: Próprio Autor.

3.1.2. Delimitação do Problema

O contexto necessário para examinar os desafios associados à especificação de requisitos em projetos de desenvolvimento ágil de *software* encontra-se na seção 1.1 deste trabalho. Embora os métodos ágeis tenham melhorado alguns aspectos do desenvolvimento de *software*, ainda persistem desafios significativos (INAYAT et al., 2015).

Nesse cenário, a integração dos métodos ATDD e PBB surge como uma possível solução para mitigar problemas e aprimorar aspectos como a especificação e validação de requisitos. No entanto, a literatura atual carece de estudos que avaliem a eficácia dessas metodologias em ambientes reais. Assim, esta pesquisa tem como objetivo tanto implementar a integração do ATDD e PBB em um contexto prático quanto avaliar sua eficácia e impacto nas atividades de desenvolvimento e na percepção da equipe envolvida.

Conforme delineado na seção 1.2.2, a questão central desta pesquisa é: “Como as práticas ATDD e PBB podem ser utilizadas de forma integrada e sistemática como estratégia para especificar requisitos em projetos ágeis, atendendo a equipes de desenvolvimento e qualidade?”. Isso nos leva a investigar os benefícios e limitações potenciais da integração entre ATDD e PBB, com o objetivo de oferecer contribuições relevantes tanto para as práticas de engenharia de software ágil quanto para o conhecimento acadêmico na área.

A implantação e avaliação do processo de integração do ATDD e PBB são descritas em detalhes nos capítulos 5 e 6, respectivamente. Para esta fase, a questão de pesquisa que orienta a análise é: “Qual é o estado atual das práticas de especificação de requisitos e colaboração entre as equipes de

desenvolvimento e qualidade na empresa, e quais lacunas ou desafios precisam ser compreendidos antes da adoção integrada do ATDD e PBB?”.

3.1.3. Definição dos Critérios de Avaliação da Pesquisa-Ação

Segundo Thiollent (2007), uma das primeiras tarefas é definir os critérios que serão utilizados para avaliar se as ações planejadas e implementadas tiveram sucesso após cada ciclo do processo de pesquisa.

Para garantir a eficácia da pesquisa-ação em nosso estudo, estabelecemos critérios de avaliação claros e objetivos. Esses critérios são essenciais para verificar se a pesquisa-ação atingiu suas metas de promover entendimento, gerar e assimilar conhecimento, e proporcionar a contribuição esperada:

- **Primeiro Critério:** Verificar se a pesquisa-ação facilitou um entendimento comum do projeto entre todos os participantes, incluindo a equipe de desenvolvimento e os *stakeholders*. Essa análise foi conduzida através de entrevistas semiestruturadas e observações diretas, com ênfase na clareza dos objetivos do projeto, na eficácia da comunicação interna e na compreensão dos requisitos para a integração do ATDD e PBB.
- **Segundo Critério:** Avaliar se a pesquisa-ação foi eficaz na criação de novo conhecimento e na facilitação de sua assimilação pelos participantes. Para isso, examinamos a produção de artefatos e a aplicabilidade das soluções desenvolvidas, observando como o conhecimento foi aplicado na prática e a habilidade da equipe em adotar novas práticas e soluções.
- **Terceiro Critério:** Avaliar o impacto da pesquisa-ação no campo do desenvolvimento ágil de software, com foco na especificação de requisitos. Esse critério foi analisado através do feedback dos participantes sobre a utilidade e aplicabilidade do processo de integração do ATDD e PBB, bem como da aderência das práticas implementadas às melhores práticas recomendadas.

Estes critérios de avaliação visam não apenas compreender os efeitos imediatos da implementação do processo de integração do ATDD e PBB em um projeto ágil específico, mas também avaliar seu impacto a longo prazo e sua relevância para a comunidade de desenvolvimento de software ágil. Os resultados da avaliação desses critérios estão apresentados na seção de Resultados da Avaliação dos Critérios de Pesquisa-Ação deste trabalho.

3.2. Procedimentos para Coleta, Análise e Síntese dos Dados

Esta seção descreve os procedimentos que foram adotados para coletar, analisar e sintetizar os dados utilizados na avaliação da situação da empresa no início da pesquisa-ação, bem como na avaliação feita após a implantação e utilização do processo de integração do ATDD e PBB.

Nesta pesquisa fez-se a utilização de dados primários, tendo sido utilizadas entrevistas semiestruturadas e observações.

Para guiar as entrevistas, foi elaborado um roteiro conforme apresentado no Apêndice A. Esse roteiro é composto por 25 questões destinadas a todos os membros da equipe e mais 10 questões específicas para o Gerente de Projetos, considerando suas atribuições no contexto organizacional.

O roteiro inicia com questões gerais e progride para outras mais específicas. Com o objetivo de validar se as questões a serem respondidas pelos entrevistados, realizou-se um estudo piloto com um profissional que atua como QA (*Quality Assurance*) e possui mais de oito anos de experiência na área de testes de *software*.

A realização da entrevista piloto permitiu identificar pontos de melhoria no instrumento de coleta de dados. Como resultado da calibração, foram realizados os seguintes ajustes: (i) reformulação de questões com redação ambígua, tornando-as mais claras e objetivas; (ii) reorganização da sequência das perguntas, visando melhorar o fluxo da entrevista e evitar redundâncias; e (iii) inclusão de perguntas de aprofundamento, como “por quê?”, com o objetivo de incentivar respostas mais detalhadas e compreender melhor as justificativas dos participantes. Esses ajustes contribuíram para aumentar a consistência e a qualidade do instrumento.

Embora o estudo piloto tenha sido conduzido com apenas um participante, essa abordagem foi considerada adequada para os objetivos desta etapa, uma vez que o foco principal foi validar a compreensão das questões e não realizar inferências estatísticas. Além disso, o participante selecionado possuía experiência significativa na área de engenharia de requisitos e testes de *software*, o que possibilitou uma avaliação crítica e qualificada do instrumento.

Após obter a autorização da empresa para entrar em contato com cada potencial participante (conforme Anexo A – Termo de Consentimento da Pesquisa), foi estabelecido um contato inicial via e-mail, convidando-os para participar do estudo. As entrevistas foram agendadas e realizadas individualmente por meio da ferramenta *Google Meet*². Inicialmente foram fornecidas aos participantes informações e orientações relevantes sobre a entrevista, o objetivo do estudo esclarecido e conscientizado ao entrevistado a importância de detalhar as respostas o máximo possível.

Na primeira fase da coleta de dados, além do registro das respostas, também foram anotadas observações complementares feitas pelos entrevistados, tais como esclarecimentos adicionais, exemplos espontâneos e comentários relevantes para a interpretação das respostas.

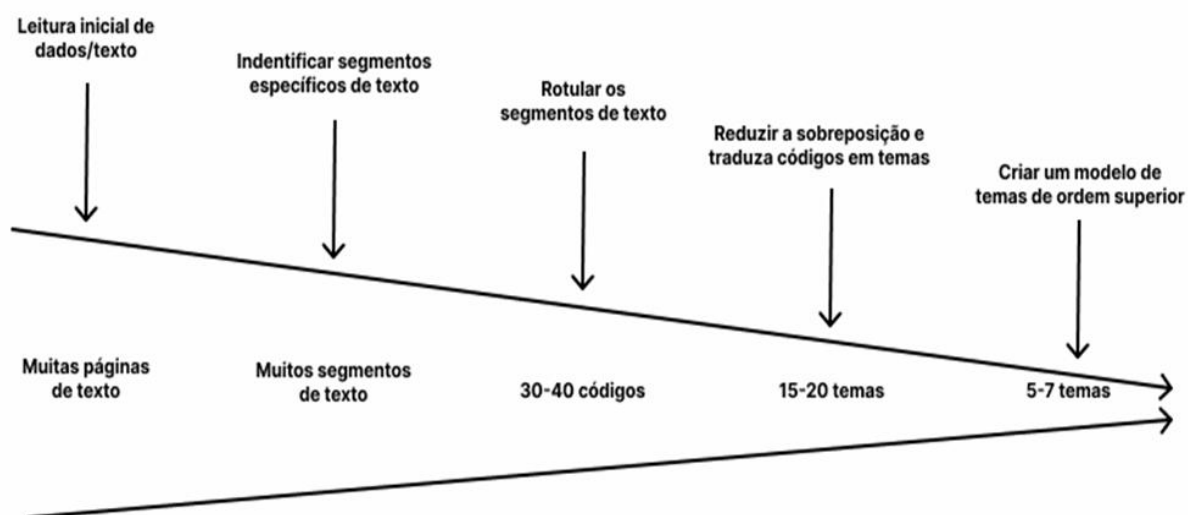
Com o objetivo de obter as informações necessárias e ao mesmo tempo ter a atenção focada durante o diálogo com os participantes, as entrevistas foram gravadas mediante autorização prévia feita no início da entrevista.

² <https://meet.google.com>

Para análise e síntese dos dados, também foi utilizado o método de síntese temática, que consiste em organizar e examinar os dados por meio da identificação e agrupamento de temas e padrões recorrentes. Este processo seguiu as diretrizes estabelecidas por Cruzes e Dyba (2011), conforme ilustrado na Figura 7.

Em resumo, o procedimento começa com uma leitura inicial dos dados coletados, seguida pela identificação e codificação de segmentos específicos. Esses códigos são então agrupados em temas, que por sua vez são organizados em categorias mais amplas. Esse processo permite construir um modelo interpretativo capaz de responder tanto à questão central da pesquisa — que investiga como as práticas ATDD e PBB podem ser utilizadas de forma integrada e sistemática — quanto à questão específica desta etapa, que busca compreender o estado atual das práticas de especificação de requisitos e da colaboração entre desenvolvimento e qualidade na empresa antes da adoção integrada do ATDD e PBB.

Figura 7 - Processo de síntese temática



Fonte: Adaptado de Cruzes e Dyba (2011).

As transcrições das entrevistas foram realizadas em duas etapas. Na primeira foi utilizada a ferramenta *Good Tape*³, na qual os áudios eram inseridos e a saída era a transcrição em texto das entrevistas. Na segunda etapa foi feito um refinamento das transcrições geradas automaticamente pela ferramenta, quando foi comparado o áudio com o texto e corrigido os erros encontrados.

Na etapa seguinte, a codificação foi aberta, sem uso de ferramentas, identificando e rotulando as palavras-chave nas respostas dos participantes de acordo com cada pergunta do roteiro. Os códigos decorrentes de cada entrevista foram constantemente comparados a códigos na mesma entrevista e de outras entrevistas e assim sendo agrupados em temas. Nesta fase, foram identificados códigos que

³ <https://goodtape.io/>

representam conceitos, ideias e padrões recorrentes nas respostas dos desenvolvedores. Alguns exemplos de códigos obtidos: “Informação mais Objetiva”, “Conhecimento do Produto”, “Informações sem ambiguidade”.

Os códigos identificados foram agrupados em temas relevantes. Por exemplo, os códigos relacionados à informação mais objetiva e informações sem ambiguidade foram agrupados no tema "Clareza". Isso destacou a importância da clareza nas informações para o time de desenvolvimento.

Os temas como "Requisitos não identificados" e "Falta de entendimento compartilhado do produto" surgiram de forma recorrente nas entrevistas iniciais, destacando aspectos essenciais na avaliação do estado inicial do projeto.

Os temas identificados foram agrupados em categorias mais amplas, como "Objetivo" e "Organização", oferecendo uma visão clara das percepções da equipe sobre o projeto e a aplicação da abordagem.

3.3. Resultados da Avaliação

Segundo Miguel et al. (2012), na pesquisa-ação, é essencial que o processo de pesquisa seja gerido de maneira proativa. Ele argumenta que a qualidade dos resultados pode depender tanto da gestão do projeto quanto da execução do próprio projeto e da análise dos resultados. Assim, a avaliação dos resultados deve ser fundamentada nos objetivos da pesquisa, sejam eles científicos ou técnicos, e nas proposições ou hipóteses formuladas no início do estudo.

De acordo com Coughlan e Coughlan (2002), a avaliação é um processo que inclui a reflexão sobre os resultados da ação, sejam eles intencionais ou não. Ela também envolve uma revisão do processo para que o próximo ciclo de planejamento e execução possa se beneficiar das lições aprendidas. A avaliação é crucial para o aprendizado; sem ela, as ações são executadas de forma aleatória, independentemente dos resultados, o que pode levar à multiplicação de erros, aumento da ineficácia e maior frustração.

Nesta seção, são expostos os resultados detalhados da avaliação conduzida no âmbito da pesquisa-ação. O foco está na análise das práticas atuais e nas percepções dos participantes. Os dados foram obtidos por meio de entrevistas semiestruturadas e observação direta, com o intuito de identificar tanto os aspectos positivos e eficazes quanto os desafios encontrados no ambiente de desenvolvimento ágil de software.

3.4.1. Avaliação do Processo de Engenharia de Requisitos

A coleta de dados durante a avaliação inicial da empresa apontou que os artefatos produzidos eram: histórias de usuário, protótipos de interface, plano de teste e os cenários de testes. As histórias eram descritas dentro do PBI (*Product Backlog Item*). Os casos de testes e o plano de testes eram

escritos no meio da *sprint*. As histórias de usuário, o plano de teste e os casos de testes eram feitos na ferramenta de gerenciamento de projeto *Azure DevOps*⁴, como mostra nas Figuras 8 e 9.

Figura 8 - História de Usuário no PBI

PRODUCT BACKLOG ITEM 10597

10597 Edição do chamado | Regra de obrigatoriedade de campos de categorização

No one selected 0 Comments Front-end

State: Done Area: Talkture

Reason: Moved out of state Testini Iteration: Talkture\Release 03\Sprint 06

Description

Eu como suporte preciso que ao editar um chamado ele obrigue o usuário a categorizar o chamado para facilitar no entendimento e direcionamento da demanda

1. Cenário Edição de Chamado

R1. Ao realizar a edição de um chamado no perfil do Suporte, tornar os campos obrigatórios para preenchimento:

- Prioridade
- Tipo do Chamado
- Software
- Categoria

R2. Caso o usuário tente salvar o chamado, deve apresentar a informação de que os campos citados na R1 precisam estar preenchidos

R3. Em tela esses campos devem estar representados com "*" no momento de abertura da edição

R4. Essa PBI não deve impactar no cenário de Abertura de chamado

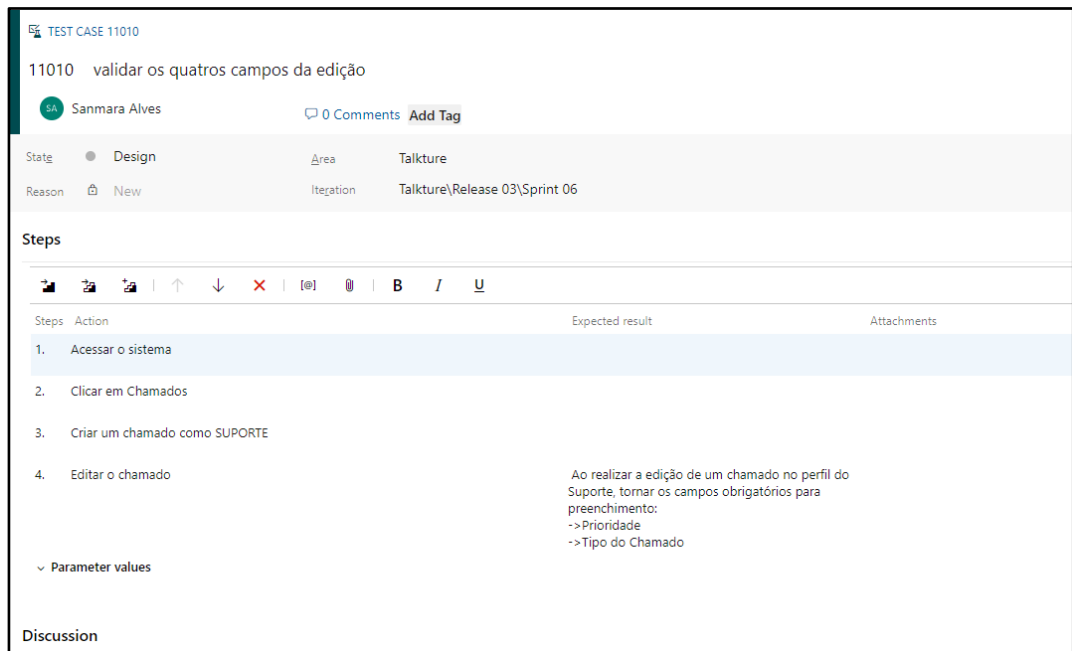
Acceptance Criteria

Click to add Acceptance Criteria.

Fonte: Próprio Autor.

⁴ <https://azure.microsoft.com/pt-br/products/devops>

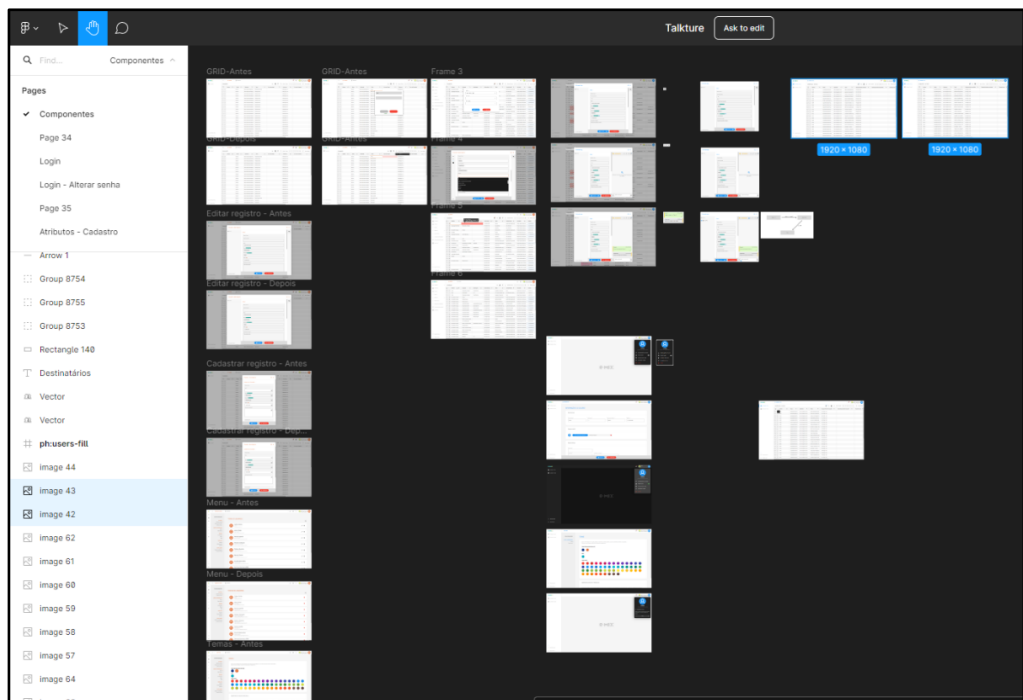
Figura 9 - Casos de Testes



Fonte: Próprio Autor.

Os protótipos eram produzidos na ferramenta *Figma*⁵, como mostrado na Figura 10.

Figura 10 - Protótipos no Figma



Fonte: Próprio Autor.

⁵ <https://www.figma.com/pt-br/>

O PO agendava uma reunião com o cliente para conversar sobre as definições da funcionalidade que seria desenvolvida. Em seguida, a funcionalidade era dividida em várias histórias do usuário e seus respectivos protótipos para que os desenvolvedores e os analistas de qualidade pudessem criar suas atividades.

Os requisitos eram detalhados e codificados de forma iterativa e incremental a cada *Sprint*, de acordo com as prioridades definidas pelo cliente. Não era adotada a técnica do PBB para a construção do *backlog*.

A empresa tinha uma equipe de Qualidade especializada para os testes. Os Analistas de Qualidade consideravam que com a história de usuário era insuficiente para a execução dos testes, era necessário elaborar outros artefatos como: Casos de Testes e Plano de Testes. Ao final de cada *sprint*, as versões parciais eram testadas internamente e posteriormente liberadas para o cliente. O processo para liberar uma versão para o cliente exigia quatro ambientes: local, teste, apresentação e produção. As validações do software eram feitas de forma remota.

3.4.2. Pontos Positivos Identificados

Durante as entrevistas semiestruturadas, foram identificados diversos pontos positivos relacionados às práticas adotadas no projeto. Ao todo, participaram desta etapa 7 profissionais, conforme caracterização apresentada anteriormente.

A análise das respostas permitiu identificar a frequência com que cada aspecto foi mencionado pelos participantes, possibilitando uma visão quantitativa complementar aos dados qualitativos. A Tabela 10 apresenta a consolidação desses resultados.

Tabela 10 - Pontos Positivos Identificados nas Entrevistas

Item Avaliado	Nº de Participantes	Classificação Predominante
Uso do <i>Azure DevOps</i>	6 de 7	Positivo
Uso do Figma para prototipação	5 de 7	Positivo
Utilização de casos de teste	6 de 7	Positivo
Participação dos <i>stakeholders</i>	5 de 7	Positivo
Uso de histórias de usuário	7 de 7	Positivo

Fonte: Próprio Autor.

De forma geral, a maioria dos participantes destacou aspectos positivos nas práticas atuais. Observou-se que 100% dos entrevistados reconheceram a importância das histórias de usuário como mecanismo de apoio à comunicação e transferência de conhecimento. Da mesma forma, o uso do *Azure*

DevOps e dos casos de teste foi amplamente citado como fator relevante para organização e qualidade do desenvolvimento.

O uso do Figma e a participação dos stakeholders também foram apontados como positivos pela maioria dos participantes, especialmente no que se refere ao entendimento das funcionalidades e alinhamento das expectativas.

Não foram identificadas avaliações predominantemente negativas nesses aspectos; entretanto, alguns participantes apresentaram percepções neutras, principalmente relacionadas ao uso de ferramentas, indicando oportunidades de melhoria na padronização e utilização dessas práticas.

3.4.3. Principais Problemas Identificados

No decorrer das entrevistas semiestruturadas realizadas com os **7 participantes**, foram identificadas diversas dificuldades que impactam negativamente o processo de desenvolvimento de software. A Tabela 11 apresenta a frequência com que cada problema foi mencionado pelos entrevistados.

Tabela 11 - Frequência dos Problemas Identificados na Fase Inicial

Problema identificado	Nº de Participantes	Percentual
Histórias de usuário sem critérios de aceitação	6	85,7%
Ausência de BDD no PBI	5	71,4%
Erros identificados apenas na apresentação ao cliente	5	71,4%
Falta de refinamento após criação da história	6	85,7%
Problemas na priorização do backlog	4	57,1%

Fonte: Próprio Autor.

Com base nesses dados, observa-se que a maioria dos participantes relatou problemas relacionados à falta de critérios de aceitação e ausência de refinamento adequado, evidenciando lacunas importantes na especificação de requisitos.

Um dos pontos críticos identificados foi a ausência do uso de BDD (*Behavior Driven Development*) nos PBIs. O BDD é uma abordagem que descreve o comportamento esperado do sistema por meio de cenários estruturados no formato *Dado-Quando-Então*, facilitando o entendimento comum entre desenvolvedores, testadores e *stakeholders*. A falta dessa prática foi apontada por 71,4% dos participantes, dificultando a criação de testes e aumentando o risco de interpretações equivocadas.

Além disso, 71,4% dos entrevistados relataram que erros são frequentemente identificados apenas na fase de apresentação ao cliente, indicando falhas na validação antecipada dos requisitos. Esse cenário está diretamente relacionado à ausência de critérios de aceitação bem definidos e à falta de alinhamento entre os membros da equipe.

Outro problema relevante, citado por 85,7% dos participantes, refere-se à ausência de refinamento contínuo das histórias de usuário após sua criação. Essa prática resulta em histórias incompletas, inconsistentes e sujeitas a retrabalho durante o desenvolvimento.

Por fim, dificuldades na priorização dos itens do *backlog* também foram mencionadas por 57,1% dos participantes, impactando diretamente o planejamento das *sprints* e a previsibilidade das entregas.

De modo geral, os problemas identificados evidenciam fragilidades no processo de Engenharia de Requisitos da empresa, especialmente no que se refere à clareza, validação e organização dos requisitos, reforçando a necessidade de adoção de práticas mais estruturadas, como a integração entre ATDD e PBB proposta neste trabalho.

3.4.4. Oportunidades de Melhoria Identificadas

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

- Incluir os Critérios de Aceitação na História do Usuário: Há uma necessidade clara de ter os critérios de aceitação incluídos na documentação, para evitar mal-entendidos e reduzir o tempo gasto na interpretação.
- Consolidação do BDD na etapa de refinamento visando os critérios de aceitação: A proposta consiste em consolidar as informações durante o refinamento. Nesse momento, os cenários serão escritos a partir da história de usuário e de seus critérios de aceitação. Esses cenários têm o papel de exemplificar os comportamentos esperados de uma funcionalidade para que ela atinja um objetivo real e agregue valor ao negócio. Dessa forma, fornecem suporte ao time para que os casos de teste automatizados sejam elaborados de acordo com os cenários descritos.
- Melhoria na construção e no refinamento do *backlog* do produto: De forma colaborativa, construir um entendimento e levar todos os envolvidos a compreensão do produto e ajudar na preparação do *backlog* para o time trabalhar de modo ágil e eficaz.

Os resultados da avaliação inicial da empresa foram considerados durante a definição do processo de integração das práticas ATDD e PBB que é apresentado no capítulo 4.

4. PROCESSO PARA INTEGRAÇÃO DAS PRÁTICAS ATDD e PBB

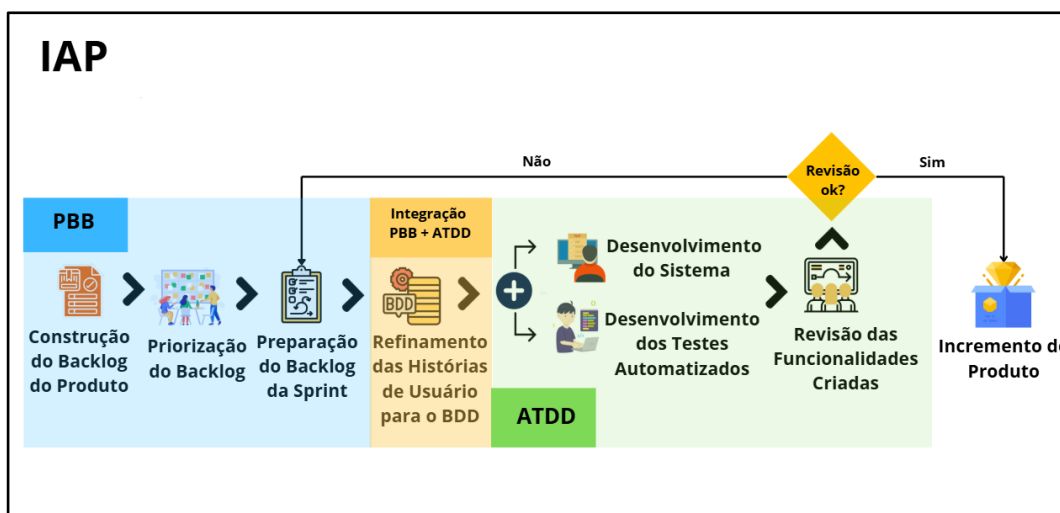
Este capítulo apresenta um processo que, através da integração das práticas ágeis ATDD e PBB, se propõe a contribuir nas atividades de requisitos e testes, e gerar um entendimento compartilhado das necessidades do produto com todo o time, através da sistematização da priorização do *backlog*, refinamento dos requisitos e da concepção da história do usuário com critérios de aceites. O processo denominado Integração de ATDD e PBB (IAP), foi proposto visando atuar nas lacunas abordadas na literatura no que diz respeito à especificação de requisitos em projetos ágeis, tendo sido considerado também o resultado da avaliação sobre o processo de desenvolvimento de uma empresa de *software*.

4.1. Visão geral do Processo para Integração de ATDD e PBB (IAP)

O processo para Integração de ATDD e PBB (IAP) foi concebido de forma a atuar na lacuna apresentada na literatura no que diz respeito às atividades de requisitos e testes em projetos ágeis, conforme mencionado na seção 1.1. Além disso, também consideramos os resultados obtidos com a avaliação realizada na empresa, conforme detalhado no capítulo 3.

O processo proposto tem como diferencial a colaboração das práticas ATDD e PBB para a especificação de requisitos desde a primeira *sprint*, podendo ser utilizado em conjunto com qualquer metodologia ágil, e adaptado para se adequar às particularidades de cada empresa. Inicialmente, a prática PBB é utilizada na concepção da história do usuário e no segundo momento, a prática ATDD atua no refinamento dessa história de usuário, visando facilitar o entendimento do desenvolvedor e demais envolvidos. A Figura 11 apresenta uma visão geral do processo IAP proposto, detalhando os subprocessos envolvidos (etapas).

Figura 11 - Visão Geral do IAP



Fonte: Próprio Autor.

O IAP inicia com a etapa “**Construção do Backlog do Produto**”, onde através da ferramenta PBB Canvas, o time pode se inserir de forma colaborativa no levantamento dos requisitos começando pelos problemas informados pelo cliente, compreendendo suas necessidades, e buscando alinhar suas expectativas como solução para o problema. Uma vez que sejam identificados eventuais problemas, as ações e os benefícios para cada perfil, é possível obter então, uma lista de funcionalidades e seus PBIs (*Product Backlog Item*).

Em seguida, deve-se realizar a etapa “**Priorização do Backlog**” através da lista de PBIs, geradas da primeira etapa do processo. Nesta atividade, o objetivo é priorizar os itens de trabalho (PBIs) para decidir em que ordem eles serão desenvolvidos.

Uma vez que os PBIs iniciais estejam definidos, organizados e priorizados, deve seguir para a etapa “**Preparação do Backlog da Sprint**”, que visa detalhar as histórias de usuário e os critérios de aceitação dos PBIs escolhidos para a *Sprint*.

Com o backlog da *Sprint* preparado, o próximo passo é a etapa “**Refinamento das Histórias de Usuários para o BDD**” que visa escrever os testes de aceitação a partir dos critérios de aceitação. Os testes devem ser escritos na linguagem adequada ao *framework* de automação de testes utilizado pelo time.

Finalizada a etapa “**Refinamento das Histórias de Usuários para o BDD**”, o processo estabelece que o time siga com duas etapas em paralelo no fluxo do processo: a) a etapa “**Desenvolvimento do Sistema**”, que a partir das funcionalidades e comportamentos mapeados na etapa anterior, tem por objetivo codificar e aplicar teste unitário em cada funcionalidade criada; b) a etapa “**Desenvolvimento dos Testes Automatizados**” tem por objetivo construir os testes de aceitação no *framework* de automação.

A última etapa “**Revisão das Funcionalidades Criadas**” consiste em verificar através das evidências gerada pelos testes, se a aplicação está atendendo aos critérios de aceitação.

Caso a etapa de revisão não apresente nenhuma não-conformidade entre os critérios de aceitação e a implementação, a *Sprint está apta a ser* entregue, ocasionando o incremento do produto. Caso contrário, novos cenários devem ser criados, e a retrospectiva acontece para promover a melhoria contínua do processo, através de ações definidas pelo time e o ciclo do IAP se reinicia para estes novos cenários.

4.2. Detalhamento do Processo

As seções a seguir detalham cada etapa do processo.

4.2.1. Construção do Backlog do Produto

Para construção do Backlog do Produto, o processo propõe a utilização da prática ágil PBB. Para isso, deve-se utilizar o Canvas como ferramenta de facilitação por prover um fluxo simples e de fácil compreensão, o que facilita o entendimento da necessidade do cliente e a construção do *backlog*, e pode ser baixado facilmente no próprio site do PBB ⁶.

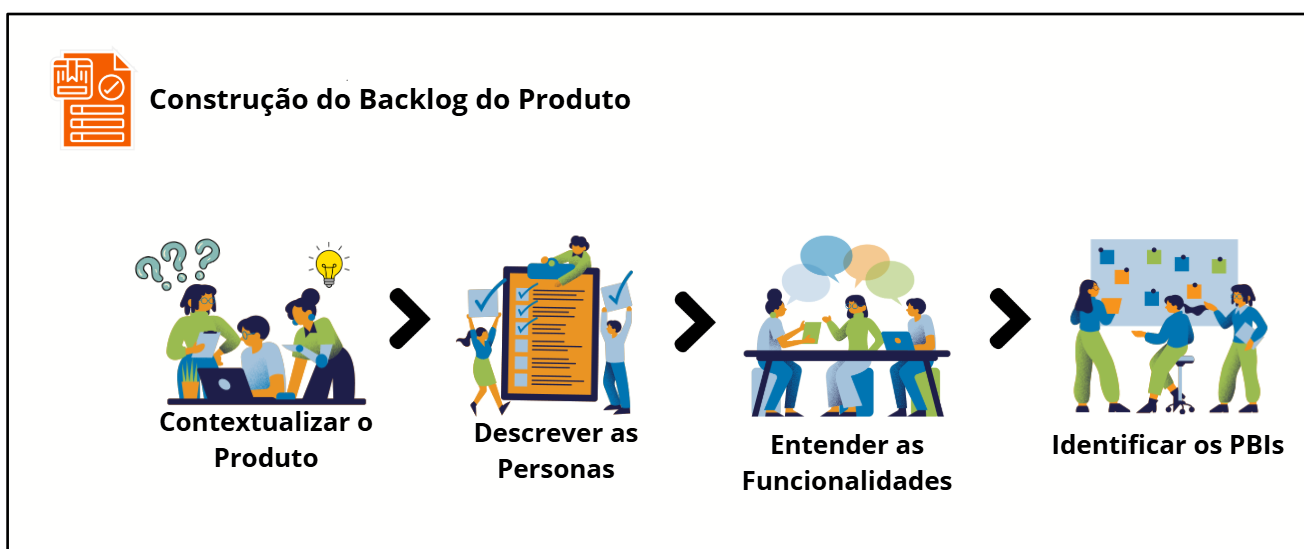
O PBB Canvas é dividido em 6 partes: *Product Name*, *Problems*, *Expectations*, *Personas*, *Features* e PBIs (*Product Backlog Item*), conforme detalhado anteriormente na seção 2.4.1. O preenchimento da ferramenta deve ser feito de forma colaborativa envolvendo pessoas de produto, negócio e o time de desenvolvimento. Entretanto, outros *stakeholders* podem ser envolvidos caso sejam considerados essenciais para um melhor entendimento do produto.

Para o PBB Canvas ser construído, e assim criar o *Backlog* do Produto, deve-se seguir as seguintes atividades (AGUIAR e CAROLI, 2021):

- Contextualizar o Produto;
- Descrever as Personas;
- Entender as Funcionalidades;
- Identificar os PBIs.

A Figura 12 apresenta a etapa (subprocesso) de Construção do *Backlog* do Produto com o fluxo das atividades apresentadas acima.

Figura 12 – Atividades do Subprocesso “Construção do *Backlog* do Produto”



Fonte: Próprio Autor.

⁶ https://www.productbacklogbuilding.com/canvas/PBB_Canvas.pdf

4.2.1.1 Contextualizar o Produto

Esta atividade tem por objetivo identificar o público-alvo, os problemas que o produto pretende resolver e suas principais funcionalidades. No início do desenvolvimento de um novo produto, é comum o surgimento de muitas dúvidas. Para que seja possível executar as etapas seguintes de desenvolvimento e testes, torna-se essencial esclarecer e compreender o que é o produto, quais problemas e necessidades ele aborda, assim como os objetivos e expectativas desejadas.

O *backlog* deve ser construído através do preenchimento do PBB Canvas. Como mostrado na Figura 13, inicia-se pelo “*Product Name*”, sendo os participantes instruídos a nomeá-lo da seguinte forma: “Imagine que este produto esteja em uma caixa, qual é o nome escrito na caixa?”.

Figura 13 - Bloco “*Product Name*” no PBB Canvas

O diagrama mostra o layout do PBB Canvas. No topo, há o título "PRODUCT BACKLOG BUILDING" e "PBB Canvas". À direita, há um campo retangular rotulado "PRODUCT NAME" que está destacado com uma borda vermelha. Abaixo, o canvas é dividido em uma grade de 2x2. O eixo vertical à esquerda contém os rótulos "PROBLEMS" (na parte superior) e "EXPECTATIONS" (na parte inferior). O eixo horizontal superior contém os rótulos "PERSONAS" (na parte superior) e "FEATURES" (na parte inferior). O eixo horizontal inferior contém o rótulo "PBI: PRODUCT BACKLOG ITEMS".

Fonte: Adaptado de Aguiar e Caroli (2021).

Em seguida, o bloco “*Problems*” deve ser abordado, conforme ilustrado na Figura 14. Por meio de um debate, deve-se identificar e compreender o estado atual do produto, ou seja, entender onde o produto está em seu ciclo de vida, e os problemas, que devem ser descritos de maneira macro. É possível haver divergências nos problemas e necessidades existentes relacionados à ideia do produto. Após o debate, deve ser realizado um agrupamento por afinidades dos problemas, seguido de uma análise do que foi levantado.

Figura 14 - Bloco “Problems” no PBB Canvas

PRODUCT BACKLOG BUILDING PBB Canvas		PRODUCT NAME
PROBLEMS	PERSONAS	
	FEATURES	
EXPECTATIONS	PBI: PRODUCT BACKLOG ITEMS	

Fonte: Adaptado de Aguiar e Caroli (2021).

Na sequência, o bloco “*Expectations*” deve ser preenchido, identificando o estado desejado do produto, ou seja, deve-se descrever as expectativas relacionadas com os problemas levantados. Um problema pode estar relacionado a uma ou mais expectativas, da mesma forma, uma expectativa pode estar relacionada a um ou mais problemas. Na Figura 15 é identificado o bloco “*Expectations*”.

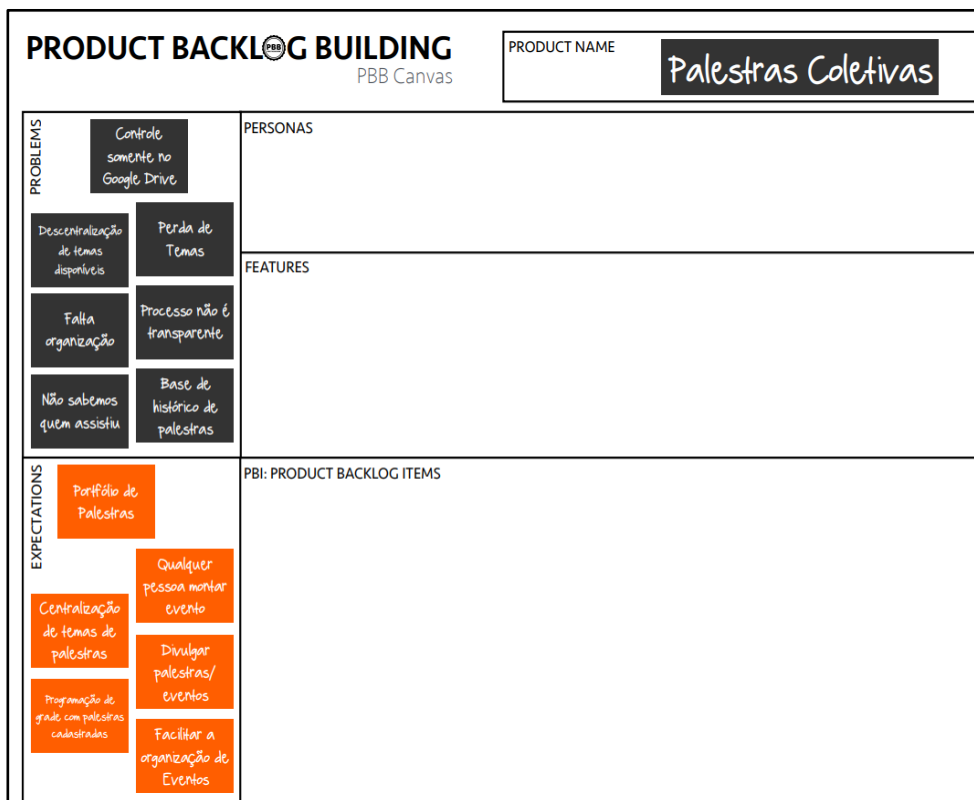
Figura 15 - Bloco do “Expectations” no PBB Canvas

PRODUCT BACKLOG BUILDING PBB Canvas		PRODUCT NAME
PROBLEMS	PERSONAS	
	FEATURES	
EXPECTATIONS	PBI: PRODUCT BACKLOG ITEMS	

Fonte: Adaptado de Aguiar e Caroli (2021).

A Figura 16 apresenta um exemplo de preenchimento dos blocos “Product Name”, “Problems” e “Expectations”, para um sistema de palestras coletivas que tem o objetivo de realizar o cadastro do palestrante, localizar o evento disponível, cadastrar o organizador do evento, entre outras funcionalidades.

Figura 16 - Exemplo Preenchimento dos blocos Product Name, Problems e Expectations no PBB Canvas



Fonte: Adaptado de Aguiar e Caroli (2021).

4.2.1.2. Descrever as Personas

Esta atividade tem o propósito de identificar e descrever quem são os usuários, papéis e responsáveis envolvidos no produto, denominados de “Persona”. Para cada “Persona”, os seguintes questionamentos devem ser feitos:

- O que eles fazem? A resposta deve ser colocada em um *post-it* à esquerda do *post-it* que representa a *Persona*.
- O que eles esperam do produto? A resposta deve ser colocada em um *post-it* à direita do *post-it* que representa a *Persona*.

Na Figura 17 é mostrado um exemplo de preenchimento do bloco *Persona*.

Figura 17 - Exemplo de Preenchimento do bloco *Persona* no PBB Canvas

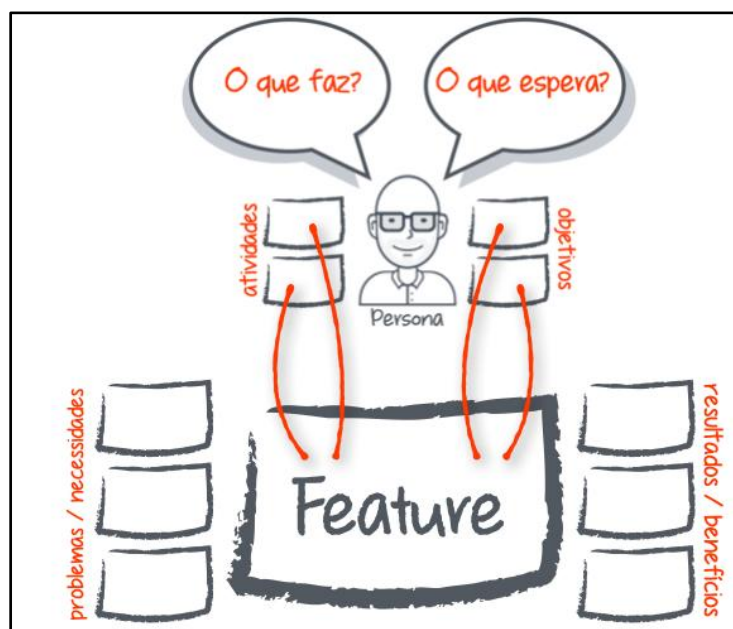


Fonte: Adaptado de Aguiar e Caroli (2021).

4.2.1.3. Entender as Funcionalidades

Funcionalidades referem-se às descrições das ações ou interações do usuário com o produto. Com esse entendimento, um *post-it* representando uma *feature*, extraído do bloco de *personas*, deve ser reescrito e transferido para o bloco de *features*. Em seguida, devem ser identificados e descritos os “Problemas” e os “Benefícios” associados a essa funcionalidade, conforme ilustrado no fluxo da Figura 18.

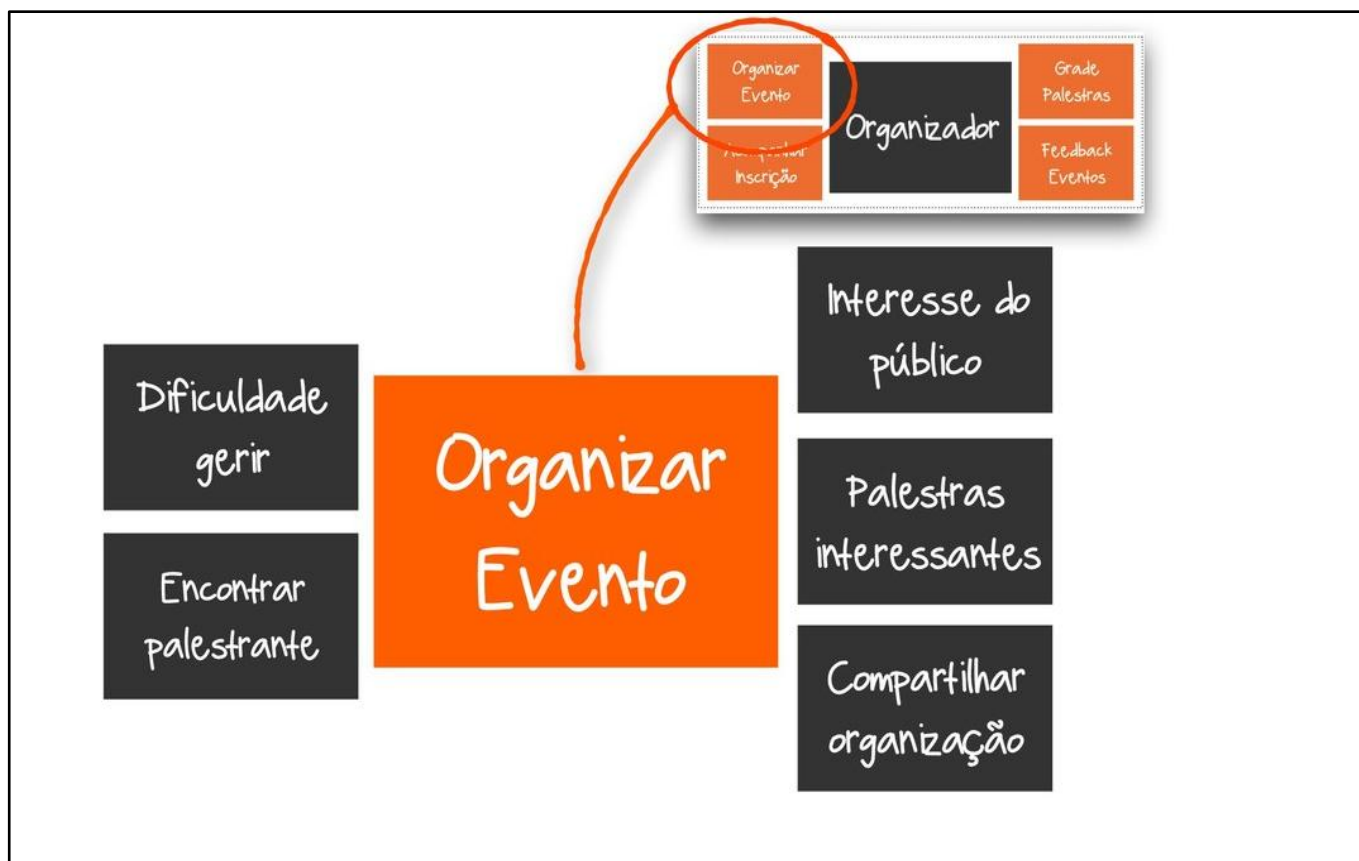
Figura 18 - Fluxo de Preenchimento da seção *Features*



Fonte: Aguiar e Caroli (2021).

Os problemas e benefícios devem ser registrados em *post-its* menores e dispostos à esquerda e à direita do *post-it* que contém a descrição da *feature*, conforme ilustrado na Figura 19. Em seguida, deve-se selecionar a próxima *feature* no bloco de *personas*, e o processo de descrição dos problemas e benefícios relacionados deve ser repetido.

Figura 19 - Exemplo do Fluxo de Preenchimento da seção *Features*



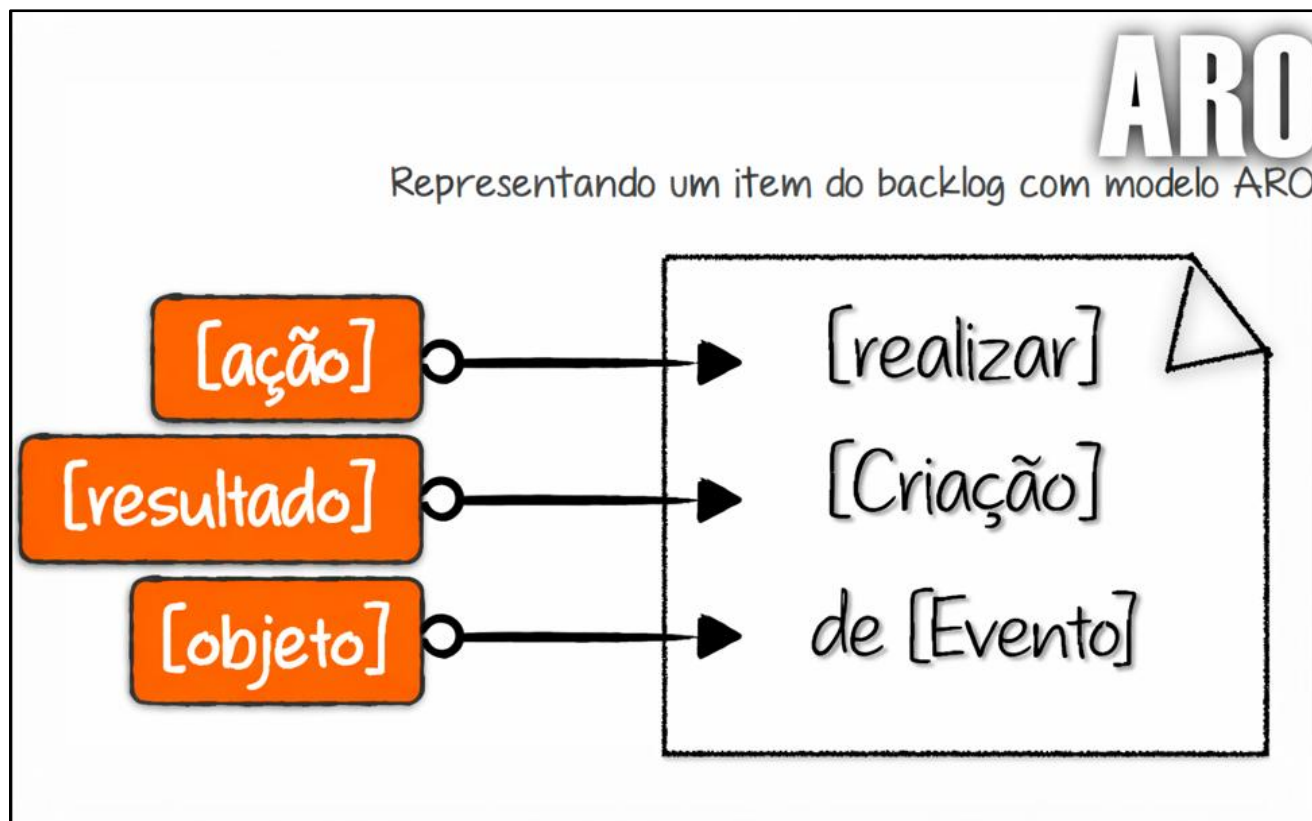
Fonte: Adaptado de Aguiar e Caroli (2021).

4.2.1.4 Identificar os PBIs

Após ter as funcionalidades bem definidas, deve-se iniciar o passo de detalhar essas funcionalidades, ou seja, a identificação dos PBIs, sendo estes identificados como ações de uma determinada *persona* no produto. Para facilitar a criação dos PBIs, deve-se fazer o seguinte questionamento aos participantes: “Qual é o primeiro item de trabalho para realizar a funcionalidade X? E o segundo? E assim por diante.

Recomenda-se a utilização do modelo desenvolvido por Peter Coad e Jeff de Luca (1997), denominado de ARO, uma representação da ação do usuário com o produto, descrito pelo acrônimo no qual o “A” é a ação, “R”, o resultado e “O”, o objeto. Na Figura 20 é possível verificar um exemplo de representação textual do modelo ARO, a representação começa com a ação (verbo), na sequência vem o resultado e termina com o objeto dentro do contexto (AGUIAR e CAROLI, 2021).

Figura 20 - Exemplo de representação textual no modelo ARO



Fonte: Adaptado de Aguiar e Caroli (2021).

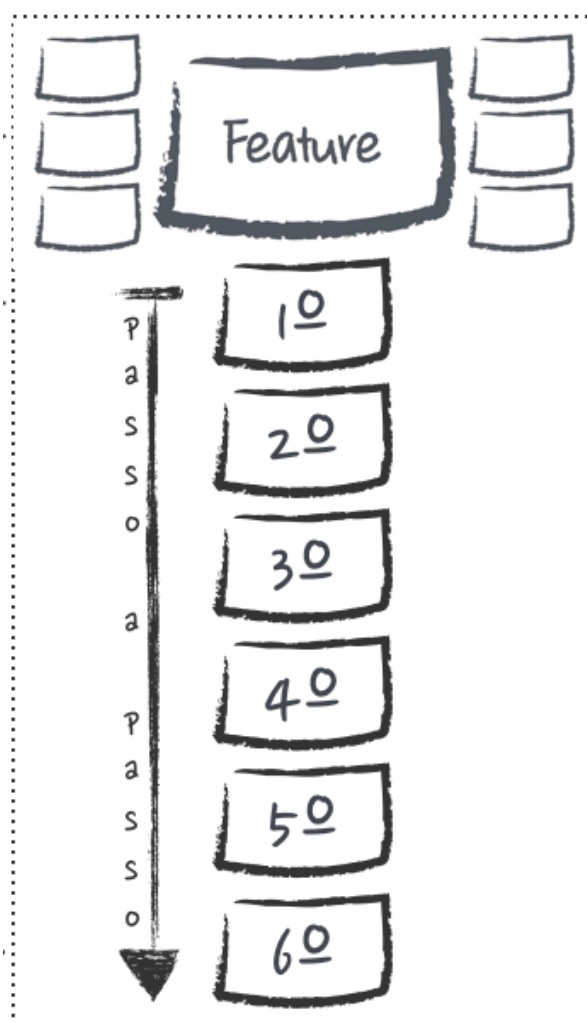
Cada PBI deve representar uma ação de um usuário no produto. Um exemplo de PBI que não está no modelo ARO: “Criar evento”. Evento criado é o resultado. Logo, deve-se buscar outro verbo para identificar a ação como, por exemplo, realizar. Reescrevendo o PBI no modelo ARO: Realizar (ação) criação (resultado) de evento (objeto).

Segundo Aguiar e Caroli (2021), a quebra de funcionalidades em PBIs é feita por meio do *Steps Map*, um método que ajuda a quebrar uma funcionalidade em pequenos passos, e cada um desses passos será um PBI.

O método é aplicado em duas etapas para descrever os PBIs no PBB Canvas (AGUIAR e CAROLI, 2021):

- **Definir o passo a passo da funcionalidade:** Deve-se questionar qual seria o primeiro item de trabalho para a implementação dessa funcionalidade e registrar a resposta em um *post-it*. Em seguida, deve-se indagar sobre o segundo item, anotando-o também em outro *post-it*, que foi colocado abaixo do primeiro. Esse processo deve ser repetido sucessivamente, e ao final, observar que cada item de trabalho representa uma etapa na construção da funcionalidade, conforme ilustrado na Figura 21.

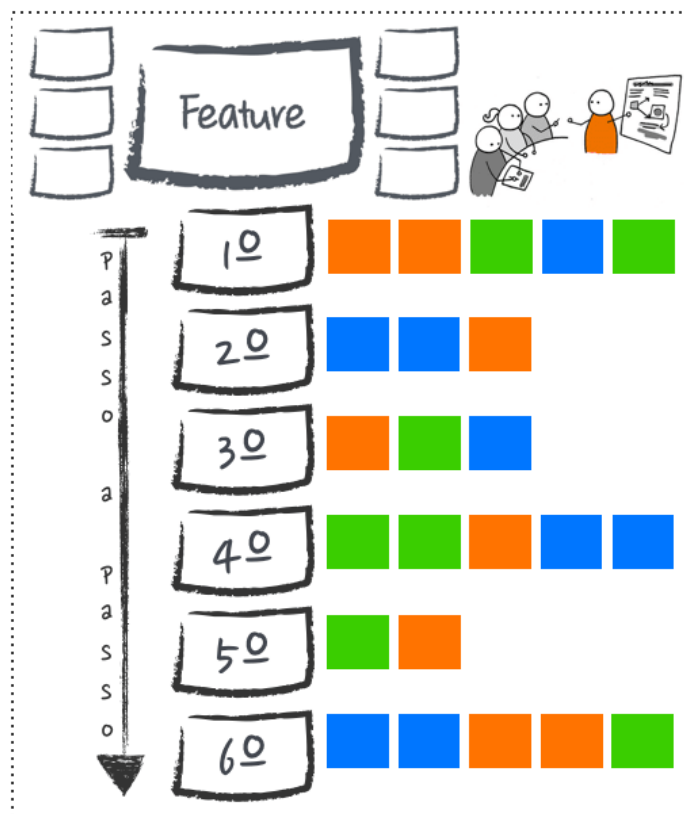
Figura 21 - Fluxo de trabalho passo a passo



Fonte: Adaptado de Aguiar e Caroli (2021).

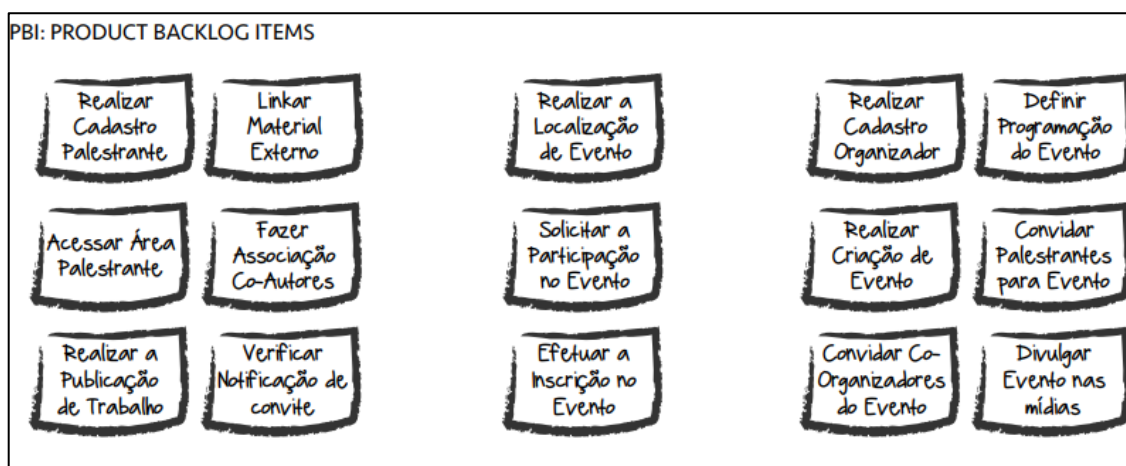
- **Evoluir cada passo com perguntas, comentários e ideias:** Após serem descritos os passos conforme mencionado na etapa anterior, é necessário reavaliar e refletir sobre cada um desses passos, conforme ilustrado na Figura 22. Os envolvidos no levantamento de requisitos devem, individualmente, registrar questionamentos, comentários e/ou ideias referentes a cada etapa em um *post-it* de cores distintas para questionamentos, comentários e ideias. Um questionamento pode eliminar um passo desnecessário. Um comentário pode aprimorar um passo já existente, e uma nova ideia pode resultar na criação de um novo passo. Ao final, conforme demonstrado na Figura 23, cada etapa do fluxo de trabalho deve ser transformada em um item do *backlog*, ou seja, um PBI (*Product Backlog Item*).

Figura 22 - Evolução com perguntas, comentários e ideias



Fonte: Adaptado de Aguiar e Caroli (2021).

Figura 23 - Itens de um Backlog do Produto



Fonte: Adaptado de Aguiar e Caroli (2021).

É importante lembrar que um *Step Maps* não é uma jornada de usuário, mas sim o passo a passo da funcionalidade até a sua completude.

A Tabela 12 resume o processo "Construção do *Backlog* do Produto" que é necessário para organizar e priorizar as funcionalidades e melhorias que direcionam o desenvolvimento de um produto. Ele descreve o propósito de garantir alinhamento com os objetivos de negócio, os pré-requisitos, como a presença de pessoas do Produto, Negócio e Desenvolvimento, as entradas necessárias, como o levantamento de requisitos, e as saídas esperadas, que incluem o preenchimento do PBB Canvas com informações detalhadas sobre o produto, *personas*, funcionalidades e PBIs. Esse processo é colaborativo e guiado pela ferramenta PBB Canvas, sendo crucial para criar uma visão clara e compartilhada do produto.

Tabela 12 - Visão Geral da Construção do *Backlog* do Produto

CONSTRUÇÃO DO <i>BACKLOG</i> DO PRODUTO	
Propósito:	É essencial para organizar, priorizar e alinhar as funcionalidades, melhorias e correções com os objetivos de negócio, garantindo que a equipe de desenvolvimento trabalhe nas tarefas que geram maior valor. Promove transparência, facilita a comunicação entre as partes interessadas e permite adaptações rápidas a mudanças, otimizando o fluxo de trabalho e o controle do progresso do projeto. Assim, o <i>backlog</i> potencializa que o desenvolvimento seja focado, eficiente e flexível, contribuindo para o sucesso contínuo do produto.
Pré-condição:	Pessoas do Produto e do Negócio estejam presentes junto com o time de desenvolvimento para dar início ao debate de ideias.
Entradas:	Levantamento de requisitos que vão surgindo no momento do debate dos blocos e que podem ser ou não considerados para entrar no PBB Canvas.
Saídas:	PBB Canvas com todos os blocos preenchidos com as informações definidas pelos envolvidos no debate. Os blocos: <i>Product Name, Problems, Expectations, Personas, Features</i> e PBIs.
Ações:	<ul style="list-style-type: none"> ● Contextualizar o Produto; ● Descrever as Personas; ● Entender as Funcionalidades; ● Identificar os PBIs.;
Ferramentas:	PBB Canvas
Papéis Envolvidos:	<ul style="list-style-type: none"> ● Pessoas do Produto e do Negócio ● Time de desenvolvimento.

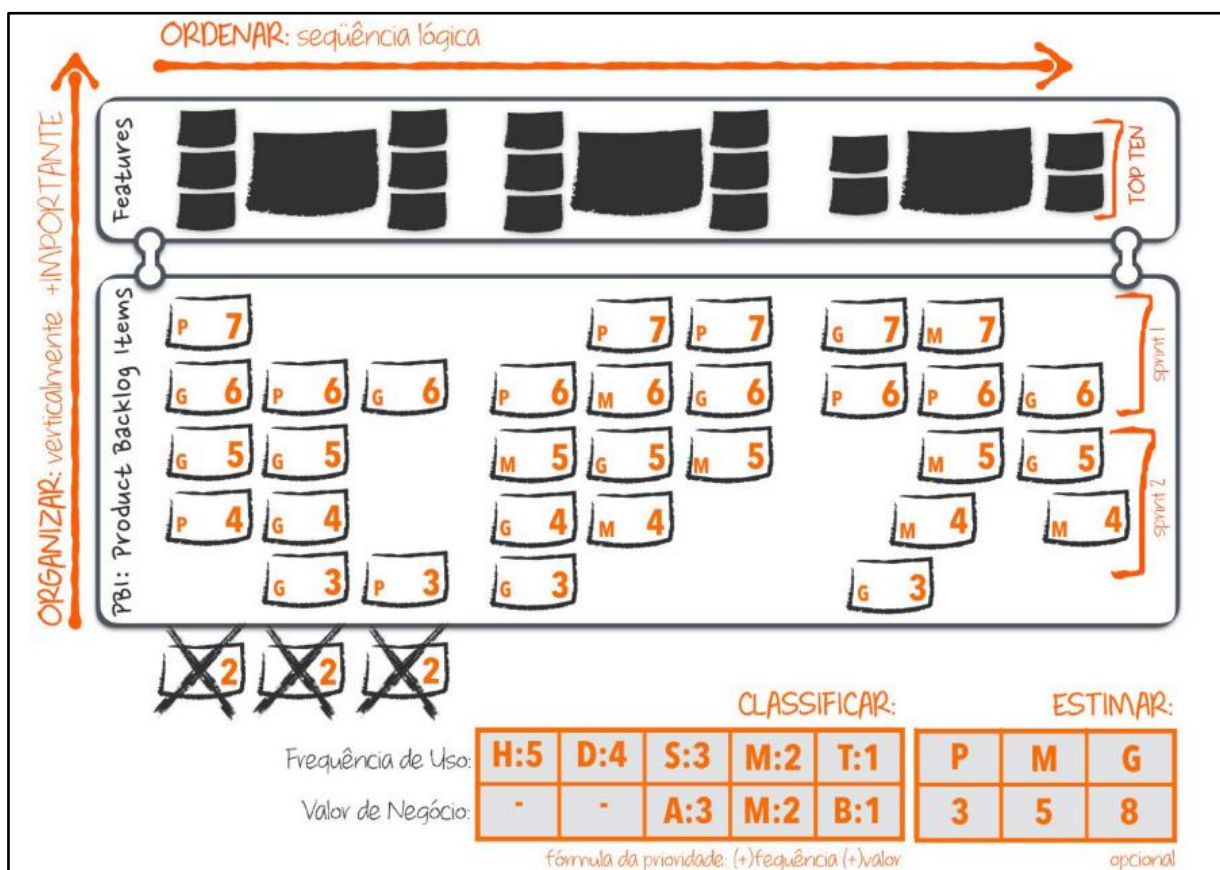
Fonte: Próprio Autor.

4.2.2. Priorização do Backlog

Esta etapa visa priorizar os itens de trabalho (PBIs) para decidir em que ordem eles serão desenvolvidos, como a metodologia utilizada neste processo é o *scrum*, os times seguem essa ordem via planejamento das *Sprints*.

A abordagem utilizada deve ser o COORG (Classificar, Ordenar e ORGANizar) do método *Product Backlog Building* (PBB) que oferece uma estrutura colaborativa e multidisciplinar para determinar a sequência de ações em um projeto. Para Aguiar e Caroli (2021), o COORG tem o objetivo de planejar e alinhar o fluxo de trabalho e/ou as próximas *Sprints*.

Figura 24 - Técnica de Priorização - COORG



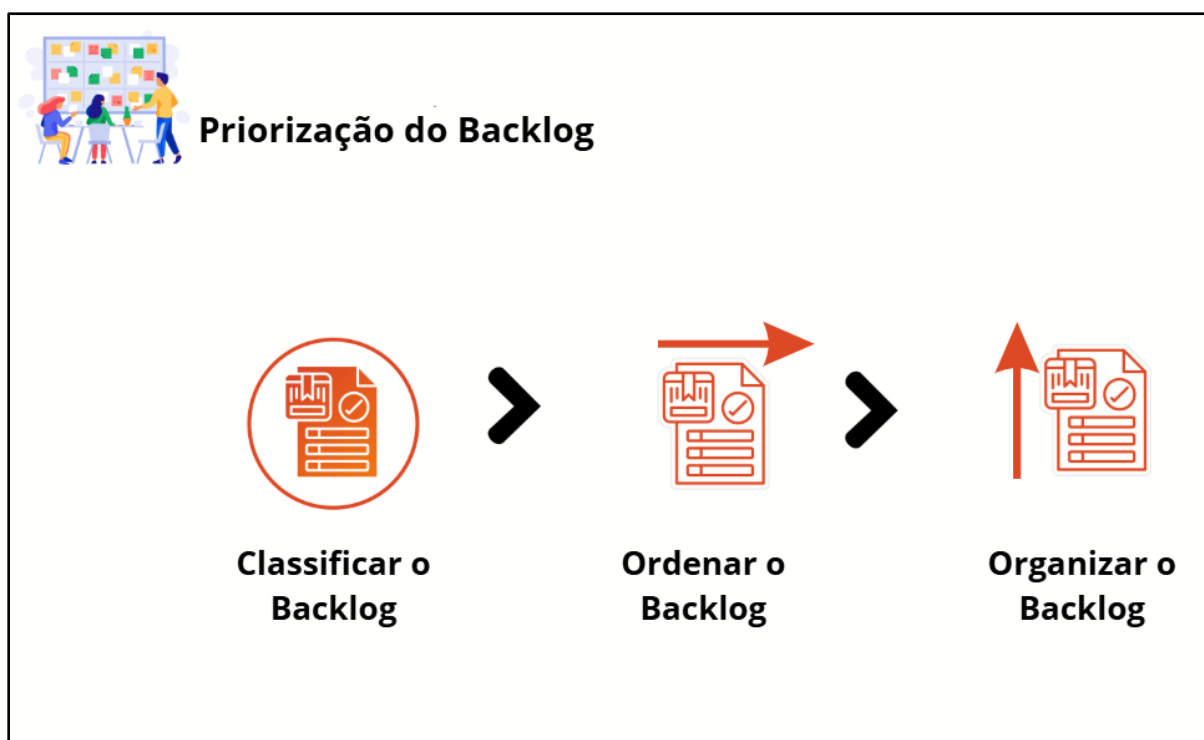
Fonte: Aguiar e Caroli (2021).

Para priorizar o backlog através do COORG, como mostrado na Figura 24, é preciso seguir os seguintes passos (AGUIAR e CAROLI, 2021):

- Classificar o *Backlog*;
- Ordenar o *Backlog*;
- Organizar o *Backlog*.

A Figura 25 apresenta a etapa de Priorização do *Backlog* com o fluxo dos passos apresentados acima.

Figura 25 - Priorização do *Backlog*



Fonte: Próprio Autor.

4.2.2.1. Classificar o *Backlog*

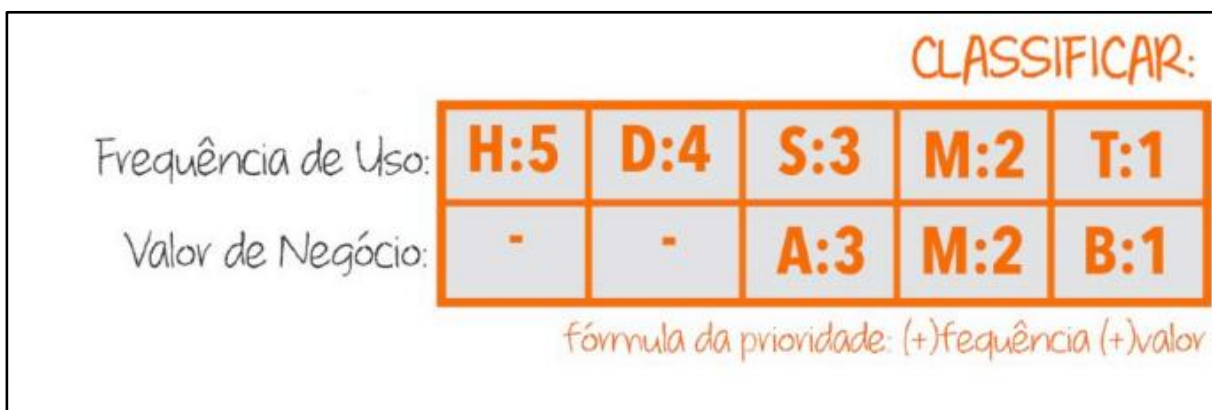
O time deve determinar quais são os critérios de classificação com suas respectivas escalas. O PBB sugere utilizar os critérios de frequência de uso e o valor de negócio, com isso, o time deve distribuir pontos nos critérios de acordo com a importância de cada PBI. Ao final, a prioridade deve se dar através da soma da frequência de uso mais o valor de negócio.

Para exemplificar a classificação, na Figura 26 apresentamos os dois critérios de classificação sugeridos pelo PBB, a saber:

- Frequência de uso: A frequência com que o usuário utiliza o PBI, foi a escala utilizada para este critério com um número escolhido pelo time entre parênteses para cada ponto:
 - Hora a Hora (5): utilizado mais de uma vez no dia.
 - Diário (4): utilizado uma vez ao dia.
 - Semanal (3): utilizado uma, duas ou três vezes na semana.
 - Mensal (2): utilizado uma vez no mês ou um pouco mais de uma vez.
 - Trimestral (1): utilizado uma vez a cada três meses.

- Valor de Negócio: O valor do negócio gerado quando o PBI é utilizado pelo usuário, foi a escala com um número escolhido pelo time entre parênteses para cada ponto:
 - Alto (3): muito importante, valor de negócio alto.
 - Médio (2): tem relevância, valor de negócio médio.
 - Baixo (1): que tem sentido, mas que não agrega muito valor no momento atual, valor de negócio baixo.

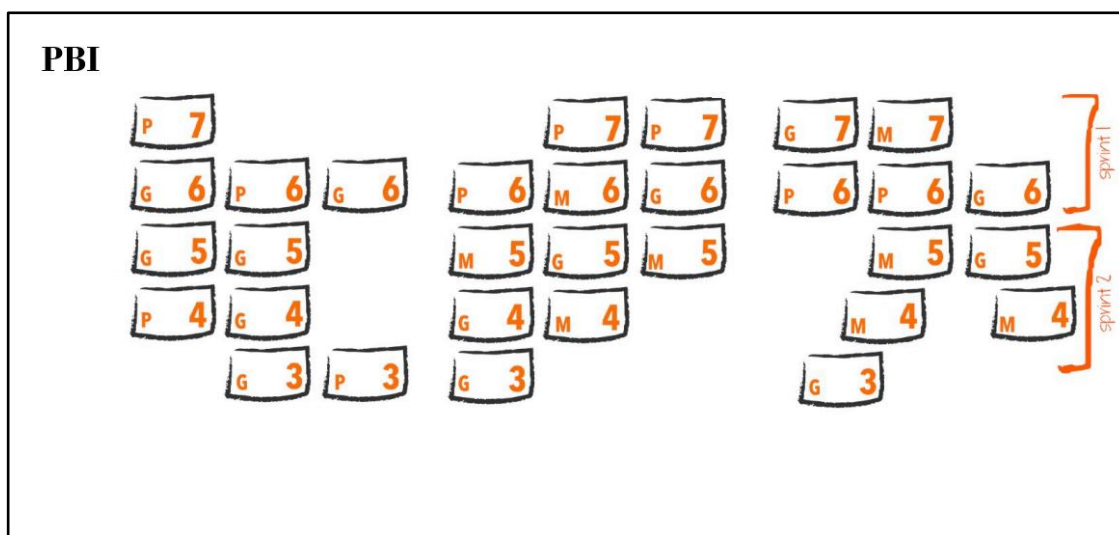
Figura 26 - Classificar o Backlog



Fonte: Aguiar e Caroli (2021).

Neste exemplo, a prioridade foi calculada a partir da seguinte fórmula: Prioridade = Frequência de uso + Valor de negócio. Por exemplo, dois PBIs receberam suas prioridades, o primeiro, por uma frequência de uso 5 (hora a hora) + um valor de negócio 1 (baixo), ficando com prioridade 6. O segundo, por uma frequência de uso 2 (mensal) + um valor de negócio 2 (médio), ficando com prioridade 4. Neste exemplo, quanto maior a pontuação, maior a prioridade para um PBI, por tanto, o primeiro PBI terá prioridade, como mostrado na Figura 27.

Figura 27 - Ordem de Prioridade

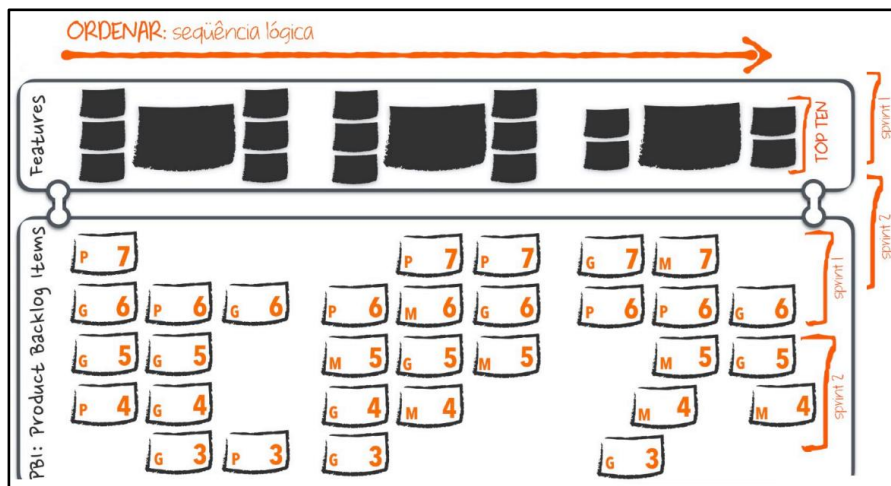


Fonte: Aguiar e Caroli (2021).

4.2.2.2. Ordenar o Backlog

A segunda etapa é ordenar as funcionalidades em uma sequência lógica, da esquerda para a direita, ao mover uma *feature*, deve-se mover também seus respectivos PBIs, colocando-os abaixo da mesma, por exemplo, o Time X prioriza as funcionalidades segundo uma jornada do usuário, então, o time deve seguir a sequência lógica em que as funcionalidades serão os passos da jornada do usuário. Na Figura 28, é possível verificar a ordenação do *backlog*.

Figura 28 - Ordenar o Backlog

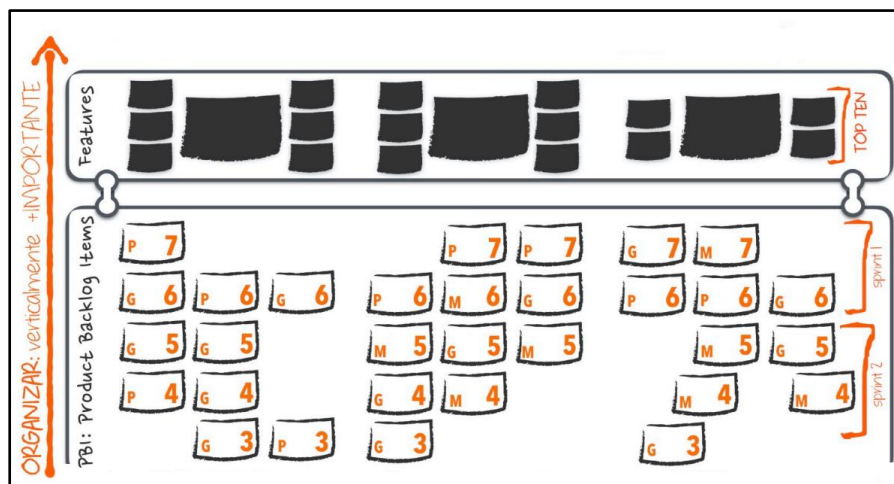


Fonte: Aguiar e Caroli (2021).

4.2.2.3. Organizar o Backlog

Para cada *feature*, deve-se organizar os PBIs de cima para baixo. Na primeira linha os PBIs com maior prioridade. Por exemplo, se o 7 for a maior pontuação, todos PBIs com nota 7 ficam na primeira linha. Abaixo dela, a linha do 6, depois do 5, e assim por diante, como mostrado na Figura 29.

Figura 29 - Organizar o Backlog



Fonte: Aguiar e Caroli (2021).

O COORG organiza os PBIs no padrão de uma matriz, começa com os itens da primeira linha, depois da segunda linha, e assim sucessivamente até o final. Geralmente os itens mais baixos não entram no *backlog*, pois fica claro o valor reduzido dentro do contexto do produto. O resultado das atividades do COORG não deve ser definitivo, onde é possível ser atualizado conforme evolução do *backlog*. (AGUIAR e CAROLI, 2021).

Uma visão geral da Priorização o *Backlog* é apresentado na Tabela 13.

Tabela 13 - Visão Geral da Priorização do Backlog

PRIORIZAÇÃO DO BACKLOG	
Propósito:	Garantir que a equipe de desenvolvimento concentre seus esforços nas funcionalidades que geram maior valor para o negócio e os usuários. Isso permite que os recursos sejam alocados de forma eficiente, potencializando que as entregas mais importantes ou urgentes sejam feitas primeiro.
Pré-condição:	PBB Canvas com todos os blocos preenchidos com as informações definidas pelos envolvidos no debate. Os blocos: <i>Product Name, Problems, Expectations, Personas, Features e PBIs</i> .
Entradas:	PBB Canvas Preenchido
Saídas:	<i>Backlog</i> alinhado e priorizado no PBB Canvas.
Ações:	<ul style="list-style-type: none"> ● Classificar o <i>Backlog</i>; ● Ordenar o <i>Backlog</i>; ● Organizar o <i>Backlog</i>.
Ferramentas:	PBB Canvas
Papéis Envolvidos	<ul style="list-style-type: none"> ● Pessoas do Produto e do Negócio ● Time de desenvolvimento.

Fonte: Próprio Autor.

4.2.3. Preparação do Backlog da Sprint

Esta etapa visa preparar o *backlog* da *Sprint*, uma vez que os PBIs iniciais já estão definidos, organizados e priorizados.

Para medir a qualidade de um PBI, é preciso verificar tanto na entrada quanto na saída de uma *Sprint*, seguindo dois conceitos fundamentais para os PBIs: definição de preparado, ou *definition of ready* (DoR) e definição de pronto, ou *definition of done* (DoD). Na preparação do Backlog, só deve entrar na *Sprint* o que estiver preparado (DoR), e só deve sair o que estiver pronto (DoD) (AGUIAR e CAROLI, 2021).

Quando o PBI está preparado para ser puxado para uma *Sprint*, isso indica que o time (AGUIAR e CAROLI, 2021):

- Tem a informação necessária para trabalhar no item;
- Entende o porquê do item;
- Consegue mostrar o término do trabalho;
- Identifica que o item compõe/se relaciona com uma funcionalidade;
- Concorda que o item cabe em uma *Sprint*.

Com base nas informações acima, o time deve verificar através do *checklist* a seguir se cada PBI atende a definição de preparado (AGUIAR e CAROLI, 2021):

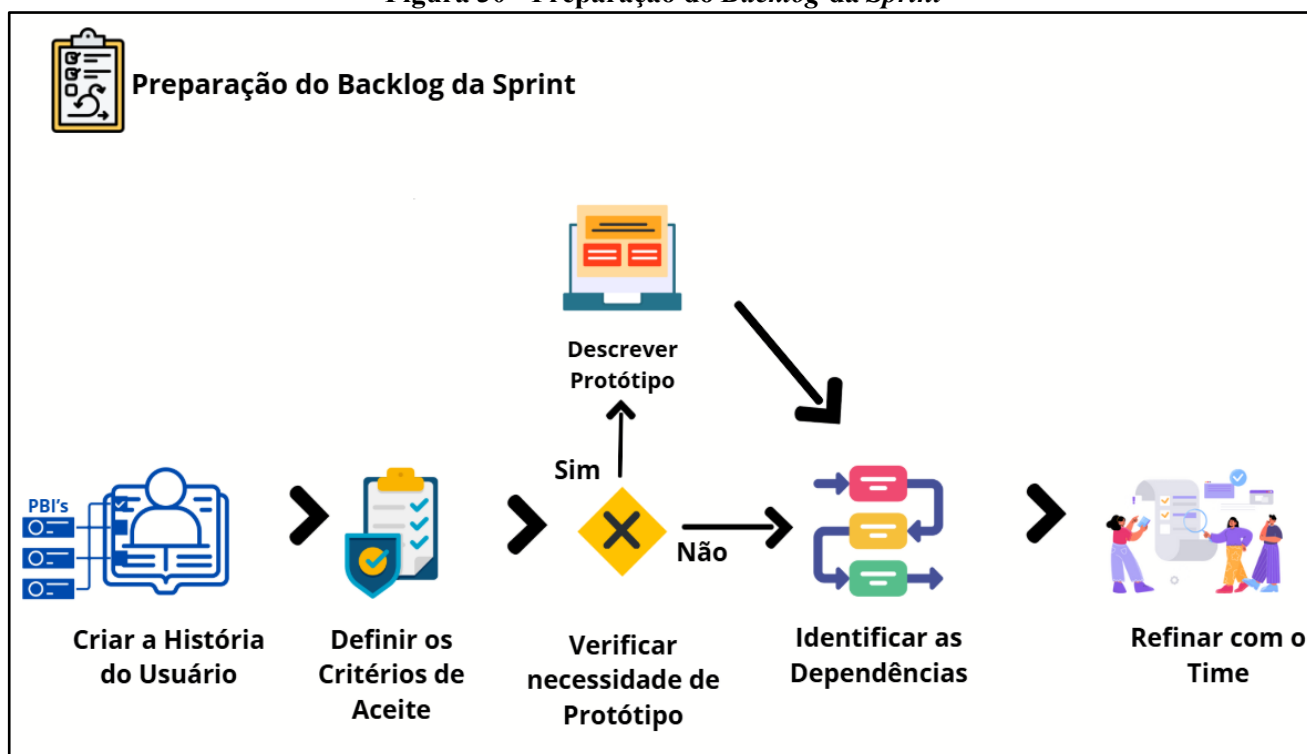
- O PBI está representado por uma história de usuário?;
- O PBI está coberto por critérios de aceite e BDD?;
- O PBI está mapeado por uma interface (quando necessário)?;
- As dependências do PBI estão identificadas (se houver)?.

Esta etapa tem como diferencial escrever uma história de usuário que contemple todos os conceitos citados acima: Critérios de aceite, BDD, Interface (quando necessário) e dependências de PBIs se houver. Mas para isso, é necessário a participação de dois papéis do time: o PO (*Product Owner*) e o QA (*Quality Analyst*).

O PO é um dos principais papéis dentro das metodologias ágeis, especialmente no *Scrum*. Ele deve ser o responsável por definir a visão do produto e garantir que o desenvolvimento esteja alinhado com os objetivos do negócio e as necessidades dos usuários. Entre suas principais funções estão gerenciar o *backlog* do produto, priorizar as tarefas, comunicar-se com as partes interessadas e potencializar que a equipe de desenvolvimento esteja trabalhando nas funcionalidades de maior valor.

O PO deve atuar como um ponto central entre o cliente e a equipe, tomando decisões sobre o que será construído, quando e com que prioridade, visando maximizar o valor do produto. O QA deve ser responsável por planejar, criar e executar testes para verificar se o software funciona corretamente, está livre de *bugs* e atende aos requisitos especificados. Além de testar o produto, o QA também deve atuar para prevenir problemas, propondo melhorias nos processos de desenvolvimento e potencializando que a qualidade seja incorporada desde as fases iniciais do projeto. O objetivo principal do QA deve ser potencializar que o produto entregue seja confiável, eficiente e atenda às expectativas dos usuários. Na Figura 30 mostra as atividades do subprocesso “Preparação do *backlog* da *sprint*”.

Figura 30 - Preparação do *Backlog* da *Sprint*



Fonte: Próprio Autor.

4.2.3.1. Criar a História de Usuário

Criar uma história de usuário de forma clara e bem estruturada é fundamental para que o time de desenvolvimento entenda o que precisa ser feito, e o QA possa validar se o que foi entregue atende aos critérios definidos. Nesta seção, será mostrado o passo a passo para o PO criar uma história de usuário com critérios de aceite, e o QA complementá-la com o BDD, conforme descrito na seção 4.2.4.

A história de usuário deve ser simples e seguir um formato padrão que ajude a entender o objetivo do usuário. Um formato comum é:

Como [usuário], eu quero [ação/desejo] para [resultado/benefício]

Exemplo: *Como um usuário logado, quero poder visualizar meu histórico de pedidos, para acompanhar o status das minhas compras anteriores.*

4.2.3.2. Definir os Critérios de Aceite

Definir os critérios de aceite é a atividade responsável por estabelecer as condições que determinam quando a história de usuário pode ser considerada concluída. Esses critérios orientam o desenvolvimento e a validação do requisito, garantindo um entendimento comum entre PO, equipe de desenvolvimento e QA. Os critérios devem ser claros, objetivos e verificáveis.

Exemplos de critérios de aceite:

- O usuário logado deve acessar o histórico de pedidos na seção “Meus Pedidos”;
- Os pedidos devem ser apresentados em ordem decrescente de data;
- Devem ser exibidos número do pedido, data, status, valor total e itens comprados;
- Deve existir a opção de filtrar pedidos por status: “Todos”, “Entregue”, “Em processamento” e “Cancelado”.

4.2.3.3. Verificar Necessidade de Protótipo

Verificar a necessidade de protótipo consiste em avaliar se a história de usuário demanda representações visuais, como *wireframes*, *mockups* ou protótipos, para apoiar o entendimento da solução proposta. Essa atividade é especialmente relevante em funcionalidades com maior complexidade visual ou interação do usuário.

4.2.3.4. Descrever Protótipo

O PO deve descrever a interface de forma textual ou anexar protótipos que ilustrem a disposição dos elementos, fluxos de navegação e comportamentos esperados da interface.

Exemplo:

- A tela “Histórico de Pedidos” deve apresentar um grid com as colunas: “Número do Pedido”, “Data”, “Status”, “Valor Total” e “Ação”;
- Deve existir um componente de filtragem por status;
- Um botão “Voltar” deve estar disponível no canto superior direito, direcionando o usuário à tela “Minha Conta”.

4.2.3.5. Identificar as Dependências

Se uma história de usuário depende de outra história ou funcionalidade para ser implementada, o PO deve identificá-las. Isso ajuda a garantir que a equipe saiba se precisa concluir outras tarefas antes de trabalhar nesta.

Exemplo:

- Esta história depende da conclusão da história "Cadastro de Pedido", que é responsável por inserir os dados de pedidos no sistema.

4.2.3.6. Refinar com o Time

Após a criação da história, o PO deve refiná-la em colaboração com o time de desenvolvimento e QA. Durante o refinamento, é importante garantir que todos entendam o propósito, os critérios de aceite e o impacto nas outras histórias.

O refinamento do *backlog* deve ser contínuo, por tanto, o dono do produto continuará trabalhando nos próximos PBIs, preparando-os para a próxima *Sprint*.

Para que o PBI fique com a definição de pronto, que é o acordo que demonstra a qualidade do PBI produzido, e que comprova a satisfação do time com o trabalho realizado, são necessários os seguintes pré-requisitos (AGUIAR e CAROLI, 2021):

- Entregar um incremento do produto;
- Contemplar os critérios de aceite estabelecidos;
- Está documentado para uso;
- Está aderente aos padrões de codificação;

Uma visão geral da atividade da Preparação do *Backlog* da *Sprint* é apresentada na Tabela 14.

Tabela 14 - Visão Geral da Preparação do *Backlog* da *Sprint*

PREPARAÇÃO DO <i>BACKLOG</i> DA <i>SPRINT</i>	
Propósito:	É essencial para garantir que a equipe de desenvolvimento tenha uma visão clara e detalhada das tarefas que precisam ser realizadas no curto prazo. Isso envolve a seleção, detalhamento e priorização de histórias de usuário ou itens do <i>backlog</i> que são viáveis de serem concluídos dentro do ciclo da <i>sprint</i> . A preparação adequada ajuda a evitar ambiguidades, potencializa que todos estejam alinhados quanto às expectativas e metas, e permite um planejamento mais preciso dos recursos e do tempo. Com um <i>backlog</i> de <i>sprint</i> bem preparado, a equipe pode trabalhar de forma mais focada, eficiente e com menos interrupções, maximizando a produtividade e a qualidade das entregas.
Pré-condição:	<i>Backlog</i> alinhado e priorizado no PBB Canvas.
Entradas:	PBIs escolhidos no PBB Canvas.
Saídas:	Template da História de usuário com Critérios de aceites disponível no Apêndice B.
Ações:	<ol style="list-style-type: none">1. O PO cria a história de usuário com a estrutura básica, incluindo:<ul style="list-style-type: none">○ Descrição clara (Como, Quero, Para)○ Critérios de aceite○ Descrição da interface (se necessário)○ Dependências de outras histórias (se houver)

	2. Após a criação da história do usuário, PO deve refiná-la em conjunto com o time e QA.
Ferramentas:	Template no Apêndice B.
Papéis Envolvidos	<ul style="list-style-type: none"> ● <i>Product Owner</i> (PO); ● QA; ● Time.

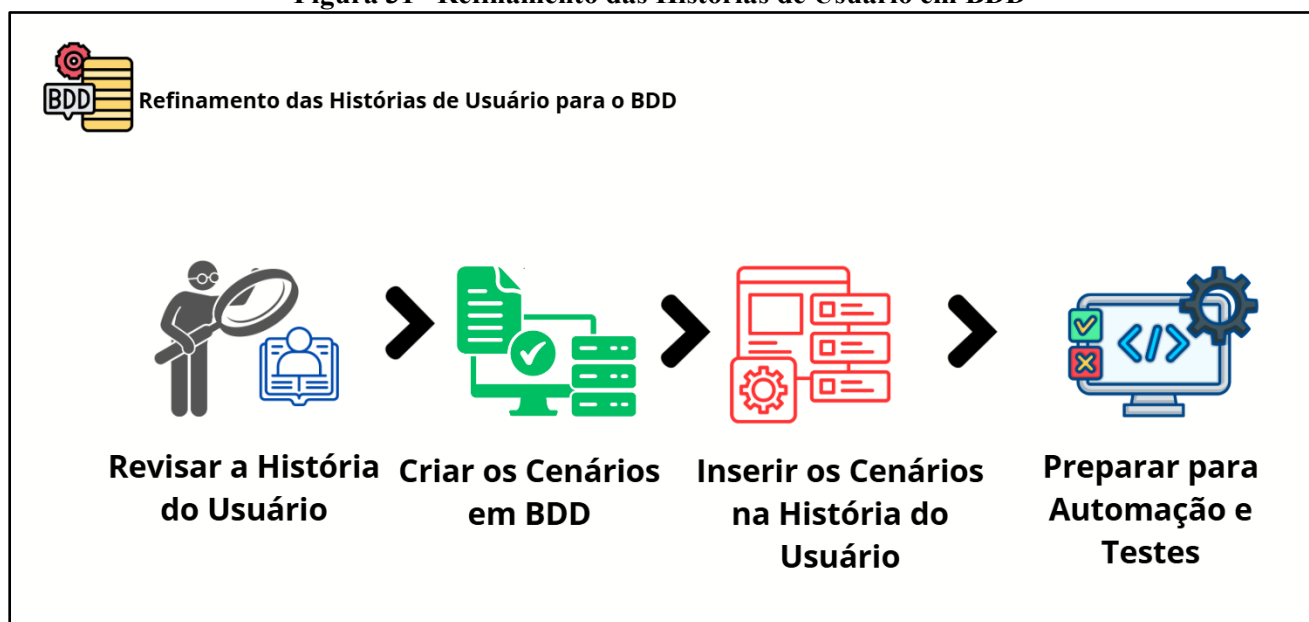
Fonte: Próprio Autor.

4.2.4 Refinamento das Histórias de Usuário para o BDD

Refinar histórias de usuário no contexto do *Behavior-Driven Development* (BDD) é uma etapa crítica para potencializar que as funcionalidades do sistema sejam descritas de forma clara, compreensível e orientada ao comportamento esperado do usuário. Esse processo envolve a transformação das histórias de usuário em cenários de teste, escritos em uma linguagem comum a todas as partes envolvidas, como desenvolvedores, testadores e *stakeholders*, utilizando o formato "Dado-Quando-Então" (MUNIZ, BOAS, CABRAL e COLARES, 2019).

Esse processo contribui para reduzir ambiguidades, melhorar a comunicação entre as equipes e garantir que o comportamento esperado do sistema seja compreendido e validado antes da implementação. A Figura 31 apresenta o fluxo do refinamento das histórias de usuário para o BDD, composto pelas atividades descritas a seguir.

Figura 31– Refinamento das Histórias de Usuário em BDD



Fonte: Próprio Autor.

4.2.4.1. Revisar a História do Usuário

Revisar a história do usuário consiste em analisar a história criada pelo *Product Owner*, verificando se o objetivo está claramente definido, se o valor de negócio está explícito e se os critérios de aceite são compreensíveis e suficientes. Essa atividade é fundamental para garantir que a história esteja pronta para ser refinada no contexto do BDD.

Para que o objetivo seja considerado claro, a história deve apresentar de forma objetiva quem é o usuário, qual ação será realizada e qual o resultado esperado, evitando descrições genéricas ou ambíguas que dificultem o entendimento da funcionalidade.

Durante a revisão, o QA e a equipe de desenvolvimento avaliam possíveis ambiguidades, inconsistências ou lacunas que possam comprometer a definição do comportamento esperado da funcionalidade. Essa atividade é realizada de forma colaborativa entre *Product Owner*, QA e equipe de desenvolvimento, caracterizando a prática de ATDD, na qual os critérios de aceitação são discutidos e alinhados antes da implementação, garantindo um entendimento compartilhado do que deve ser desenvolvido e validado.

4.2.4.2. Criar os Cenários em BDD

Criar cenários de teste em BDD consiste em transformar cada critério de aceite da história de usuário em um ou mais cenários que descrevam, de forma estruturada, o comportamento esperado do sistema. Essa atividade assegura que todos os requisitos definidos sejam compreendidos, testáveis e validados antes da implementação, contribuindo para a redução de ambiguidades e retrabalho.

Cada cenário deve representar uma situação específica a ser validada, contemplando não apenas os fluxos principais, mas também exceções e condições alternativas. Cabe ao QA garantir que todos os critérios de aceite estejam devidamente cobertos por cenários bem definidos, assegurando que o comportamento do sistema seja validado sob diferentes contextos de uso.

A criação dos cenários em BDD deve seguir o formato “Dado–Quando–Então”, que organiza a descrição do comportamento em três etapas principais. Inicialmente, deve-se identificar o cenário a ser testado, associando-o diretamente a um critério de aceite da história de usuário. Em seguida, o cenário deve ser estruturado considerando: Dado (*Given*), que define o estado inicial do sistema ou as condições prévias necessárias; Quando (*When*), que descreve a ação realizada pelo usuário ou o evento ocorrido; e Então (*Then*), que especifica o resultado esperado após a execução da ação.

Por fim, os cenários de BDD devem ser documentados diretamente na história de usuário, logo após os critérios de aceite, utilizando o padrão Dado–Quando–Então. Essa prática garante maior visibilidade dos comportamentos esperados para toda a equipe, além de assegurar que os cenários estejam claros, consistentes e alinhados aos critérios definidos pelo *Product Owner*.

4.2.4.3. Inserir os Cenários na História do Usuário

Associar os cenários de BDD à história de usuário consiste em garantir que os cenários estejam diretamente vinculados ao requisito, servindo como referência para desenvolvimento e testes. Essa associação permite que a equipe utilize os cenários como base para a implementação da funcionalidade e para a criação de testes automatizados.

Ao inserir os cenários na história, a equipe de desenvolvimento passa a utilizar essas informações como referência para a implementação da funcionalidade, enquanto o QA pode utilizá-las como base para a definição dos testes.

Exemplo:

História de Usuário: Como um cliente, eu quero visualizar o histórico de pedidos, para acompanhar o status das minhas compras.

Critérios de Aceite:

1. O usuário deve poder visualizar o histórico de pedidos após realizar login.
2. Os pedidos devem ser exibidos em uma lista ordenada por data.
3. O sistema deve permitir filtrar pedidos por status (Entregue, Processando, Cancelado).

BDD - Cenários de Teste:

Cenário 1: Exibir o histórico de pedidos corretamente

- Dado que o usuário está logado no sistema
- E acessa a página "Meus Pedidos"
- Quando o usuário visualizar o histórico de pedidos
- Então o sistema deve exibir todos os pedidos ordenados por data de compra, do mais recente ao mais antigo.

Cenário 2: Filtrar pedidos por status "Entregue"

- Dado que o usuário está na página "Meus Pedidos"
- E há pedidos com o status "Entregue"
- Quando o usuário selecionar o filtro "Entregue"
- Então o sistema deve exibir apenas os pedidos com o status "Entregue"

Cenário 3: Histórico de pedidos sem pedidos realizados

- Dado que o usuário está logado no sistema

- E acessa a página "Meus Pedidos"
- Quando o usuário não tiver nenhum pedido realizado
- Então o sistema deve exibir a mensagem "Nenhum pedido encontrado"

O QA deve garantir que todos os critérios de aceite sejam testados com cenários bem definidos. Cada cenário BDD deve cobrir diferentes comportamentos do sistema, validando tanto o sucesso quanto falhas ou condições excepcionais.

4.2.4.4. Preparar para Automação e Testes

Preparar para automação de testes consiste em garantir que os cenários de BDD estejam escritos de forma clara, padronizada e adequada para futura automação. Para isso, os cenários devem estar descritos com passos objetivos, sem ambiguidades, utilizando uma estrutura consistente no formato “*Dado–Quando–Então*” e com linguagem compreensível tanto para a equipe técnica quanto para os *stakeholders*.

Essa atividade envolve a validação da consistência dos cenários, assegurando que não existam variações desnecessárias na escrita de passos semelhantes, a eliminação de ambiguidades que possam gerar interpretações distintas durante a automação, e a verificação da reutilização de passos comuns, de forma a facilitar a manutenção e reduzir a duplicidade nos testes automatizados.

No contexto do ATDD, os cenários de BDD servem como base para os testes de aceitação automatizados, orientando o desenvolvimento desde o início e assegurando que o comportamento esperado do sistema seja continuamente validado ao longo do ciclo de desenvolvimento (MUNIZ; BOAS; CABRAL; COLARES, 2019).

Dessa forma, o refinamento das histórias de usuário por meio da definição colaborativa dos critérios de aceitação e da construção dos cenários em BDD caracteriza a prática de ATDD neste processo, uma vez que estabelece, antes da implementação, um entendimento compartilhado do comportamento esperado do sistema entre *Product Owner*, QA e equipe de desenvolvimento.

Uma visão geral da atividade de refinar as Histórias de Usuário para o BDD é apresentada na Tabela 15.

Tabela 15 - Visão geral do Refinamento das Histórias de Usuário para o BDD

REFINAMENTO DAS HISTÓRIAS DE USUÁRIO PARA O BDD	
Propósito:	Transformar funcionalidades em cenários de teste claros e orientados ao comportamento esperado do usuário, utilizando o formato "Dado-Quando-Então". Esse refinamento detalha os critérios de aceitação, potencializando que cada história seja testável e compreensível para

	<p>todos os envolvidos, como desenvolvedores, testadores e <i>stakeholders</i>. A colaboração entre o <i>Product Owner</i>, a equipe de desenvolvimento e QA é fundamental para alinhar expectativas, identificar dependências e cobrir cenários alternativos. Esse processo aprimora a comunicação, reduz ambiguidades e facilita a automação dos testes, garantindo entregas de alta qualidade.</p>
Pré-condição:	<i>Backlog da Sprint.</i>
Entradas:	Histórias de usuários e critérios de aceitação.
Saídas:	<i>Template dos Cenários de testes em BDD disponível no Apêndice C.</i>
Ações:	<ol style="list-style-type: none"> 1. Transformação dos Critérios em Cenários de Teste: O QA revisa os critérios de aceite junto com o PO e o Time, em seguida, cria cenários de teste no formato BDD (Dado-Quando-Então) para cada critério, cobrindo diferentes comportamentos e resultados esperados. 2. Documentação dos Cenários: O QA insere esses cenários diretamente na história de usuário, garantindo a cobertura completa dos critérios
Ferramentas:	<i>Template no Apêndice C.</i>
Papéis Envolvidos	<ul style="list-style-type: none"> ● <i>Product Owner (PO);</i> ● QA; ● Time.

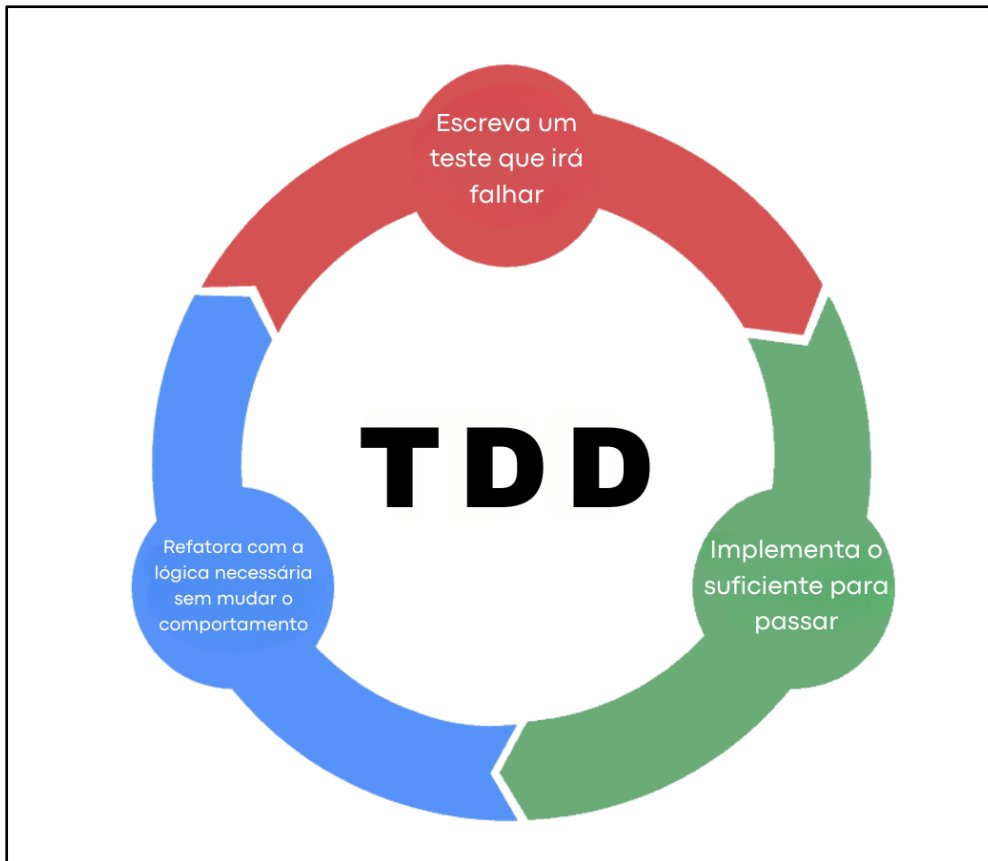
Fonte: Próprio Autor.

4.2.5. Desenvolvimento do Sistema

O desenvolvimento do sistema, no contexto deste trabalho, é orientado tanto pelos princípios do TDD quanto do ATDD. Enquanto o TDD foca na criação de testes unitários antes da implementação, o ATDD complementa esse processo ao utilizar os critérios de aceitação e cenários definidos anteriormente como base para guiar o desenvolvimento das funcionalidades. Dessa forma, a implementação não é direcionada apenas pela lógica técnica, mas principalmente pelo comportamento esperado do sistema, previamente alinhado entre os *stakeholders*.

Com a atividade de refinar histórias de usuário para o BDD finalizada, os desenvolvedores conseguem ter a visão de todas as funcionalidades mapeadas e dos comportamentos esperados delas. Nesta fase, orientado aos critérios de aceitação, o desenvolvedor criará o código através dos testes unitários seguindo o método TDD, como é mostrado na Figura 32.

Figura 32 - Sequência básica do TDD



Fonte: Adaptado de Muniz, Boas, Cabral e Colares (2019).

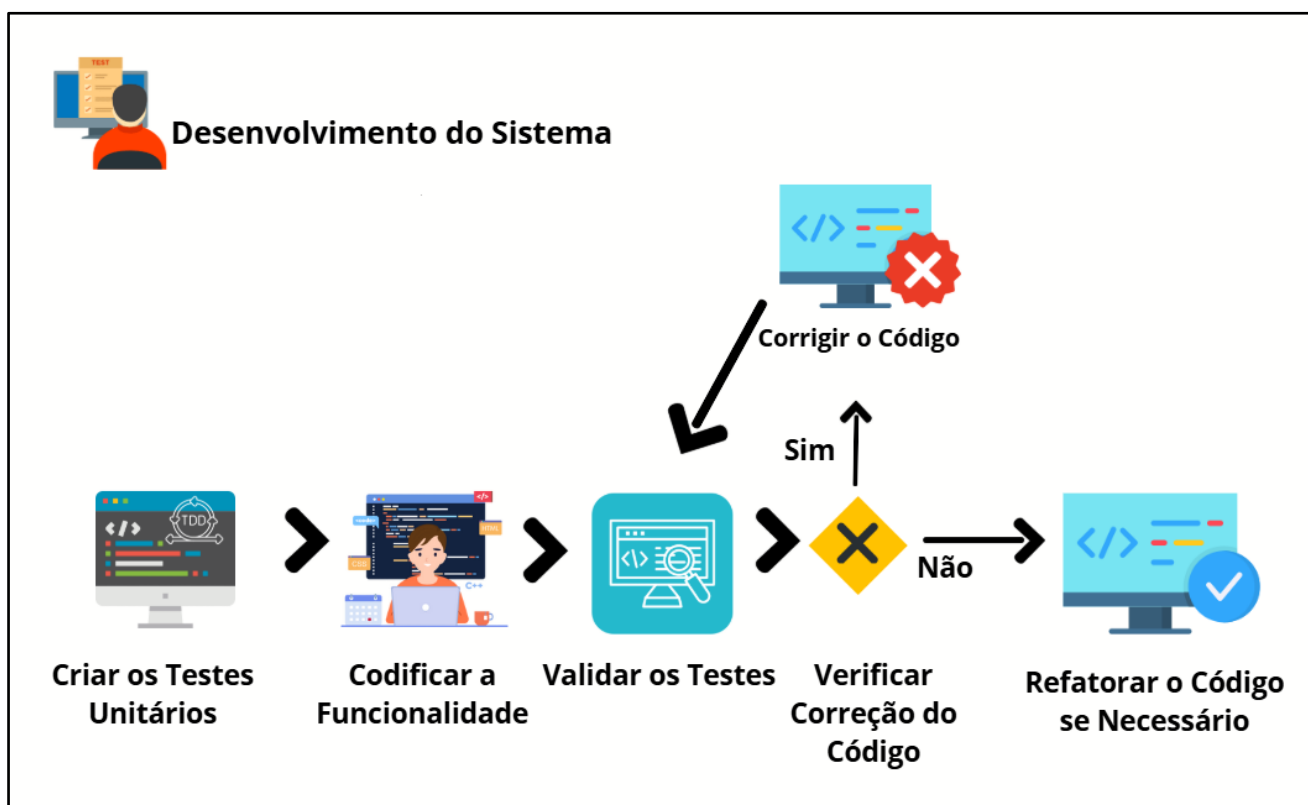
Conforme Muniz et al. (2019), a sequência básica para o TDD funciona em três passos, são eles:

- O primeiro passo (vermelho) representa um resultado negativo, o desenvolvedor deve escrever um teste que irá falhar. Essa falha acontece porque não tem código ainda, e sim um teste vazio baseado em um cenário de teste.
- No segundo passo (verde) representa um resultado positivo, o desenvolvedor escreve somente o código suficiente para passar no teste. Ao executar a compilação deste código, o teste irá passar.
- No terceiro passo (azul) representa a melhoria contínua do código, o desenvolvedor melhora o código sem alterar o seu comportamento. Este refatoramento deve considerar melhorar a qualidade da escrita do código, tais como: legibilidade, padrões e cobertura de testes unitários.

A combinação de TDD com BDD pode ser uma abordagem poderosa para garantir que o desenvolvimento esteja alinhado com os requisitos do negócio e ao mesmo tempo coberto por testes. Enquanto o TDD se concentra nos testes de unidade (ou funcionais) baseados no código, o BDD foca

na criação de cenários que descrevem o comportamento do sistema do ponto de vista do usuário ou do negócio. Na Figura 33 é possível verificar o fluxo do Desenvolvimento do Sistema.

Figura 33 - Desenvolvimento do Sistema



Fonte: Próprio Autor.

Será mostrado as atividades para implementar TDD através do BDD no subprocesso de Desenvolvimento do Sistema.

4.2.5.1. Criar os Testes Unitários

Para esta atividade, são executados dois passos:

1. Escrever os Casos de Teste

Com o TDD, os testes são escritos antes de desenvolver a funcionalidade. Esses testes são derivados dos cenários de aceitação definidos previamente no contexto do ATDD, garantindo que a implementação esteja alinhada aos comportamentos esperados do sistema. Exemplo de Testes (utilizando *frameworks* de testes JUnit):

Cenário 1 na Figura 34: Exibir o histórico de pedidos corretamente.

O Cenário 1 é um teste unitário, utilizando a anotação `@Test`, que verifica se o sistema exibe corretamente o histórico de pedidos ordenado a partir da data mais recente para a mais antiga.

Figura 34 - Exibir o histórico de pedidos corretamente

```
@Test
public void testHistoricoDePedidosOrdenadosPorData() {
    // Dado que o usuário está logado no sistema
    User user = new User("cliente", "senha123");
    user.login();

    // E acessa a página "Meus Pedidos"
    List<Order> pedidos = user.acessarHistoricoDePedidos();

    // Quando o usuário visualizar o histórico de pedidos
    // Então o sistema deve exibir todos os pedidos ordenados
    // por data de compra, do mais recente ao mais antigo.
    List<Order> pedidosOrdenados = Order.ordenarPorData(pedidos);
    assertTrue(pedidosOrdenados.equals(pedidos));
}
}
```

Fonte: Próprio Autor.

Cenário 2 na Figura 35: Filtrar pedidos por status "Entregue".

O Cenário 2 é um teste que verifica se ao filtrar os pedidos pelo status "Entregue", o sistema retorna apenas os pedidos que correspondem a esse status. Essa validação é importante para garantir que o filtro funcione corretamente.

Figura 35 - Filtrar pedidos por status

```
@Test
public void testFiltroDePedidosPorStatusEntregue() {
    // Dado que o usuário está na página "Meus Pedidos"
    User user = new User("cliente", "senha123");
    user.login();

    // E há pedidos com o status "Entregue"
    user.addPedido(new Order("Entregue", data1));
    user.addPedido(new Order("Processando", data2));

    // Quando o usuário selecionar o filtro "Entregue"
    List<Order> pedidosFiltrados = user.filtrarPedidosPorStatus("Entregue");

    // Então o sistema deve exibir apenas os pedidos com o status "Entregue"
    for (Order pedido : pedidosFiltrados) {
        assertEquals("Entregue", pedido.getStatus());
    }
}
}
```

Fonte: Próprio Autor.

Cenário 3 na Figura 36: Histórico de pedidos sem pedidos realizados.

O Cenário 3 é um teste que verifica se o sistema lida corretamente com a situação em que um usuário acessa a funcionalidade "Meus Pedidos", mas ainda não realizou nenhum pedido. É esperado que o sistema exiba uma mensagem informando que não há pedidos.

Figura 36 - Histórico de pedidos sem pedidos realizados

```
@Test
public void testHistoricoSemPedidos() {
    // Dado que o usuário está logado no sistema
    User user = new User("cliente", "senha123");
    user.login();

    // E acessa a página "Meus Pedidos"
    List<Order> pedidos = user.acessarHistoricoDePedidos();

    // Quando o usuário não tiver nenhum pedido realizado
    // Então o sistema deve exibir a mensagem "Nenhum pedido encontrado"
    assertTrue(pedidos.isEmpty());
    assertEquals("Nenhum pedido encontrado", user.getMensagem());
}
}
```

Fonte: Próprio Autor.

2. Executar os Testes

Após a elaboração dos testes, é necessário executá-los para verificar seu comportamento. Neste estágio, espera-se que todos os testes falhem, uma vez que a funcionalidade correspondente ainda não foi implementada. Este é um comportamento esperado no processo de desenvolvimento orientado a testes, pois potencializa que os testes estão corretamente configurados para validar as funcionalidades que serão desenvolvidas.

4.2.5.2. Codificar a Funcionalidade

Com os testes devidamente criados e falhando, a próxima atividade é desenvolver a funcionalidade que permita a passagem dos testes. O objetivo é implementar o código necessário para atender aos critérios estabelecidos nos testes e cenários descritos. A funcionalidade deve:

- Exibir o histórico de pedidos.
- Ordenar os pedidos por data.
- Filtrar pedidos pelo status.
- Exibir uma mensagem se não houver pedidos.

A implementação deve ser feita de maneira a cobrir todos os requisitos funcionais e comportamentais descritos.

4.2.5.3. Validar os Testes

Uma vez que a funcionalidade foi desenvolvida, deve-se proceder com a reexecução dos testes para validar a implementação. Todos os testes devem agora ser bem-sucedidos, indicando que a funcionalidade foi corretamente implementada e está em conformidade com os requisitos definidos pelos cenários de teste. Caso algum teste ainda falhe, deve-se revisar o código e ajustar a funcionalidade ou os próprios testes, conforme necessário.

4.2.5.4. Verificar Correção do Código

Quando ocorrem falhas nos testes, é necessário analisar cuidadosamente o código implementado para identificar a origem do problema. Nesta atividade, o desenvolvedor deve verificar se:

- A lógica implementada atende corretamente aos critérios de aceite;
- O comportamento descrito nos testes foi interpretado corretamente;
- Não existem erros de lógica ou omissões na implementação.

Essa verificação é essencial para garantir que o código reflita fielmente os cenários definidos no BDD.

4.2.5.5. Corrigir o Código

Após a identificação das inconsistências, o código deve ser ajustado para corrigir os problemas encontrados. As correções devem ser realizadas de forma incremental, sempre seguidas da reexecução dos testes unitários, garantindo que as alterações resolvam as falhas sem introduzir novos erros.

Esse ciclo de correção e revalidação pode se repetir até que todos os testes passem com sucesso.

4.2.5.6. Refatorar o Código se Necessário

Com todos os testes passando, inicia-se a atividade de refatoração. O objetivo é melhorar a qualidade interna do código, tornando-o mais legível, organizado e fácil de manter, sem alterar o comportamento da funcionalidade.

Após a refatoração, os testes unitários devem ser executados novamente para assegurar que as melhorias realizadas não impactaram negativamente o comportamento do sistema.

Dessa forma, o desenvolvimento da funcionalidade é orientado não apenas pelos testes unitários definidos no contexto do TDD, mas também pelos critérios de aceitação e cenários previamente estabelecidos no ATDD, garantindo que a implementação esteja alinhada ao comportamento esperado do sistema desde sua concepção.

Uma visão geral da atividade de Desenvolvimento do Sistema é apresentada na Tabela 16.

Tabela 16 - Visão geral do Desenvolvimento do Sistema

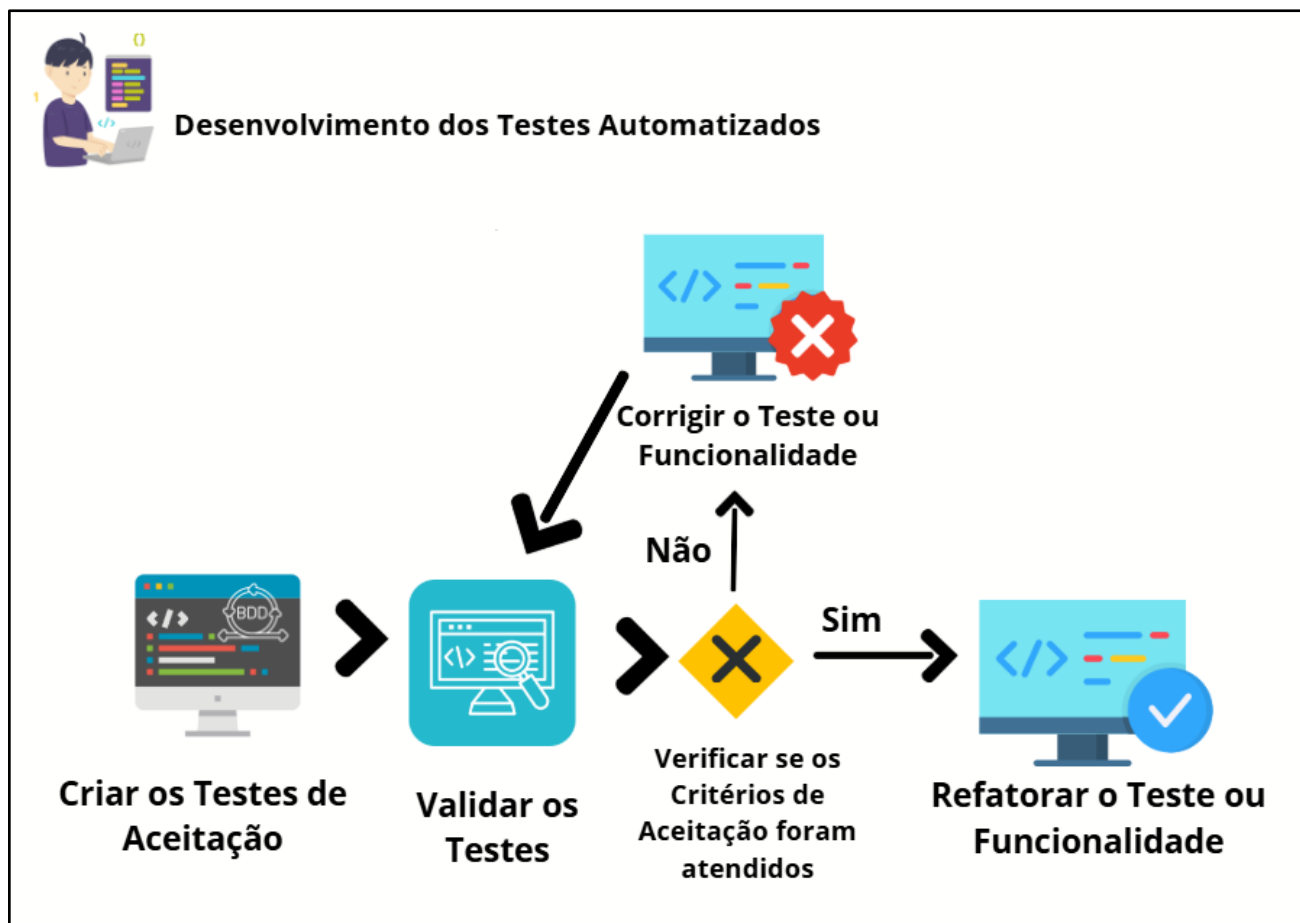
DESENVOLVER O SISTEMA	
Propósito:	No desenvolvimento, adota-se práticas como o <i>Test-Driven Development</i> (TDD), em que testes são escritos antes do código, e o <i>Behavior-Driven Development</i> (BDD), para alinhar o desenvolvimento aos requisitos de negócios. A implementação é realizada de forma incremental, com testes contínuos para validar cada componente. Após o desenvolvimento, o sistema é testado, refinado, e otimizado para garantir desempenho, segurança e usabilidade adequados.
Pré-condição:	Histórias de usuários e critérios de aceitação prontos.
Entradas:	Cenários de testes em BDD e Histórias de usuário refinadas.
Saídas:	Código-fonte das funcionalidades com o TDD implementado.
Ações:	Criação do código-fonte utilizando o método TDD com base nos cenários de testes;
Ferramentas:	Editor de código-fonte;
Papéis Envolvidos	<ul style="list-style-type: none"> ● Desenvolvedores

Fonte: Próprio Autor.

4.2.6. Desenvolvimento dos Testes Automatizados

Em paralelo com a atividade de desenvolver o sistema, o analista de qualidade deve iniciar a construção dos testes de aceitação no *framework* de automação assim que uma funcionalidade estiver pronta. A Figura 37 ilustra as atividades envolvidas no desenvolvimento dos testes automatizado.

Figura 37 - Desenvolvimento dos Testes Automatizados



Fonte: Próprio Autor.

No contexto do ATDD, esses testes são definidos antes ou durante o desenvolvimento, com base nos critérios de aceitação e cenários de BDD previamente elaborados, funcionando como guia para validação contínua do sistema. Dependendo da arquitetura da aplicação em desenvolvimento, é possível até deixar os testes de aceitação implementados mesmo com a funcionalidade ainda não desenvolvida. Os testes de aceitação irão falhar, mas assim que o desenvolvedor entregar a funcionalidade, os testes irão executar e informar o *feedback* ao desenvolvedor, caso a aplicação esteja pronta, é possível verificar a necessidade de refatoração dos testes.

4.2.6.1. Criar os Testes de Aceitação

Nesta atividade, o analista de qualidade cria os testes de aceitação automatizados com base nos cenários definidos no BDD. Cada cenário representa um comportamento esperado do sistema e é convertido em um teste automatizado no *framework* escolhido, neste trabalho, o *Robot Framework*, utilizando o conceito de BDD.

Para criar um teste automatizado no *Robot Framework* ⁷ utilizando o conceito de BDD será mostrado um passo a passo abaixo:

1: Instalar o *Robot Framework*

Para instalar o *Robot Framework*, é preciso utilizar o prompt de comando (cmd), executando o comando:

```
pip install robotframework
```

2: Configurar o ambiente de testes

Deve ser criada uma pasta para organizar os testes e dentro dela, também deverá ser criado subpastas para manter os testes em BDD organizados. No exemplo da Figura 38, existe a pasta *tests* com sua subpasta *historico_pedidos* e dentro dela o arquivo *test_historico_pedidos.robot* que é o teste em BDD.

Figura 38 - Estrutura de Pastas do *Robot*

```
/tests
  /historico_pedidos
    test_historico_pedidos.robot
```

Fonte: Próprio Autor.

3: Estrutura básica de um arquivo.robot

No arquivo *test_historico_pedidos.robot*, será adicionado a estrutura básica para começar a escrever os testes. Cada cenário do BDD será representado por uma *Keyword* dentro do *Robot Framework* como mostrado na Figura 39.

⁷ <https://robotframework.org/>

Figura 39 - Exemplo da estrutura inicial do arquivo.robot

```
*** Settings ***
Library          SeleniumLibrary # Exemplo: Para testes de UI
Suite Setup      Setup Browser
Suite Teardown   Close Browser

*** Variables ***
${URL}           https://www.seusite.com/login
${BROWSER}       Chrome

*** Test Cases ***
Exibir o histórico de pedidos corretamente
    Dado que o usuário está logado no sistema
    E acessa a página "Meus Pedidos"
    Quando o usuário visualizar o histórico de pedidos
    Então o sistema deve exibir todos os pedidos ordenados por data de compra,
    do mais recente ao mais antigo.

Filtrar pedidos por status "Entregue"
    Dado que o usuário está na página "Meus Pedidos"
    E há pedidos com o status "Entregue"
    Quando o usuário selecionar o filtro "Entregue"
    Então o sistema deve exibir apenas os pedidos com o status "Entregue"

Histórico de pedidos sem pedidos realizados
    Dado que o usuário está logado no sistema
    E acessa a página "Meus Pedidos"
    Quando o usuário não tiver nenhum pedido realizado
    Então o sistema deve exibir a mensagem "Nenhum pedido encontrado"
```

Fonte: Próprio Autor.

4: Implementar as Keywords

No bloco ***** Keywords *****, será definido como as ações descritas no BDD serão automatizadas. Será utilizado a biblioteca *SeleniumLibrary*, para poder automatizar interações com a interface do usuário. Na Figura 40, é mostrado um exemplo de implementação.

Figura 40 – Exemplo de implementação

```
*** Keywords ***
Setup Browser
    Open Browser    ${URL}    ${BROWSER}
    Maximize Browser Window

Dado que o usuário está logado no sistema
    Input Text    id=username_field    my_username
    Input Text    id=password_field    my_password
    Click Button  id=login_button
    Wait Until Page Contains Element  id=meus_pedidos_link

E acessa a página "Meus Pedidos"
    Click Link    id=meus_pedidos_link
    Wait Until Page Contains Element  id=historico_pedidos

Quando o usuário visualizar o histórico de pedidos
    # Aqui poderia incluir verificações específicas da lista de pedidos
    Wait Until Page Contains Element  id=historico_pedidos

Então o sistema deve exibir todos os pedidos ordenados por data de compra,
do mais recente ao mais antigo
    Element Should Be Visible    id=pedido_mais_recente
    # Adicione aqui validações para garantir que os pedidos estão ordenados corretamente

E há pedidos com o status "Entregue"
    Wait Until Page Contains    status=Entregue

Quando o usuário selecionar o filtro "Entregue"
    Click Element    id=filtro_entregue
    Wait Until Page Contains Element    id=lista_pedidos_entregue

Então o sistema deve exibir apenas os pedidos com o status "Entregue"
    Element Should Be Visible    id=lista_pedidos_entregue
    Element Text Should Be    id=status_pedido_1    Entregue

Quando o usuário não tiver nenhum pedido realizado
    # Simula situação onde não há pedidos
    Clear Element    id=historico_pedidos

Então o sistema deve exibir a mensagem "Nenhum pedido encontrado"
    Element Should Be Visible    id=nenhum_pedido_encontrado
```

Fonte: Próprio Autor.

4.2.6.2. Validar os Testes

Após a criação dos testes de aceitação, estes devem ser executados para validar se estão corretamente configurados. Nesta etapa, o objetivo não é validar a funcionalidade em si, mas garantir que os testes automatizados:

- Estão corretamente estruturados;

- Possuem *keywords* bem definidas;
- Executam conforme o fluxo descrito nos cenários de BDD.

Para executar os testes, deve-se navegar até a pasta onde o arquivo.robot está localizado e executar o seguinte comando:

robot test_historico_pedidos.robot

Os três cenários de teste descritos no BDD foram rodados:

- Exibir o histórico de pedidos corretamente
- Filtrar pedidos por status "Entregue"
- Histórico de pedidos sem pedidos realizados

É esperado que os testes falhem caso a funcionalidade ainda não tenha sido implementada. Esse comportamento confirma que os testes estão aptos a detectar desvios e fornecer *feedback* confiável quando a funcionalidade começar a ser desenvolvida.

4.2.6.3. Verificar se os Critérios de Aceitação foram atendidos

Com a funcionalidade implementada pelo desenvolvedor, os testes de aceitação devem ser reexecutados. Nesta atividade, verifica-se se os critérios de aceitação definidos nos cenários de BDD estão sendo atendidos pela aplicação.

Caso os testes sejam executados com sucesso, entende-se que a funcionalidade está em conformidade com os comportamentos esperados. Caso contrário, os resultados da execução indicam quais critérios não foram atendidos, permitindo identificar falhas de implementação ou inconsistências entre o que foi especificado e o que foi desenvolvido.

4.2.6.4. Corrigir o Teste ou a Funcionalidade

Quando os testes indicam falhas nos critérios de aceitação, é necessário realizar uma análise para identificar a origem do problema. Essa atividade pode resultar na correção:

- Da funcionalidade implementada, quando o comportamento do sistema não atende ao que foi especificado; ou
- Do próprio teste automatizado, quando a automação não representa corretamente o cenário descrito no BDD.

Após as correções, os testes devem ser novamente executados, garantindo que os ajustes realizados solucionaram o problema identificado. Esse ciclo de correção e reexecução reforça o *feedback* contínuo entre qualidade e desenvolvimento, conforme apresentado no fluxo da Figura 38.

4.2.6.5. Refatorar o Teste ou a Funcionalidade

Quando todos os testes de aceitação são executados com sucesso e os critérios de aceitação são atendidos, recomenda-se realizar a refatoração dos testes automatizados ou da funcionalidade, quando necessário. A refatoração tem como objetivo melhorar a legibilidade, organização e manutenção, sem alterar o comportamento validado pelos testes.

No contexto dos testes automatizados, a refatoração pode envolver a reorganização de *keywords*, reutilização de componentes e padronização da estrutura dos arquivos. Já na funcionalidade, a refatoração busca melhorar a qualidade do código-fonte. Após essa atividade, os testes devem ser executados novamente para assegurar que as melhorias realizadas não impactaram o comportamento esperado do sistema.

Dessa forma, os testes automatizados não apenas validam a funcionalidade implementada, mas também materializam a prática de ATDD, assegurando que o comportamento do sistema esteja em conformidade com os critérios de aceitação definidos colaborativamente desde as etapas iniciais do processo.

A Tabela 17 apresenta um resumo da atividade Desenvolvimento dos Testes Automatizados.

Tabela 17 - Resumo do Desenvolvimento dos Testes Automatizados

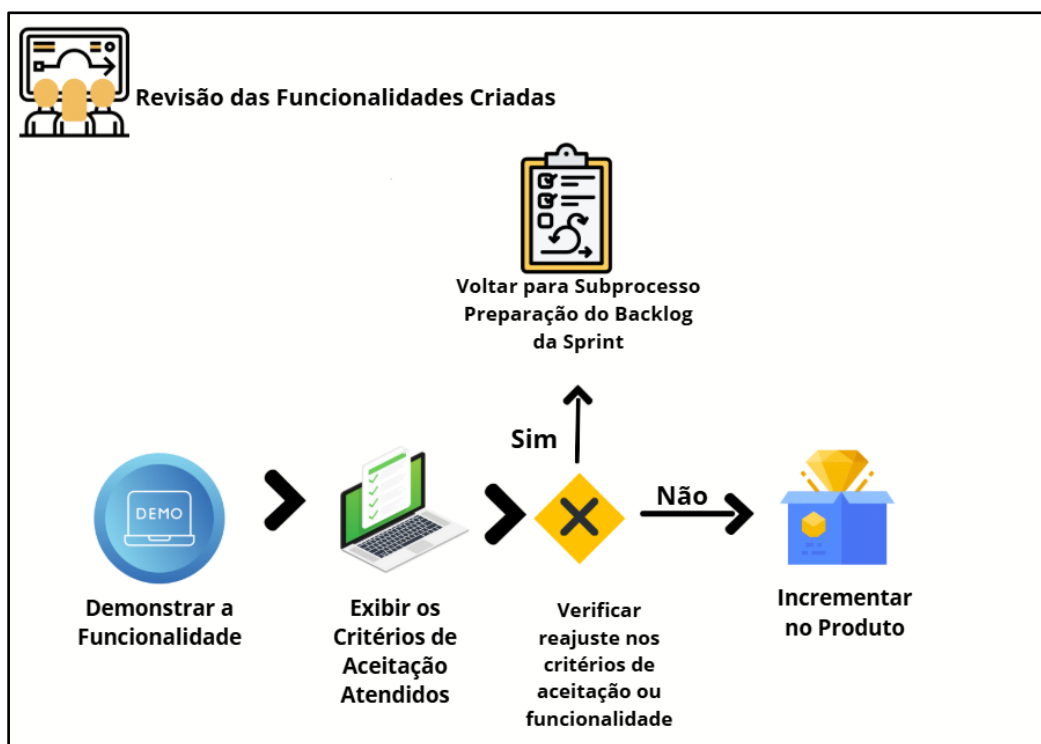
DESENVOLVER OS TESTES AUTOMATIZADOS	
Propósito:	Desenvolver testes automatizados com BDD envolve criar cenários de teste que descrevem o comportamento esperado do sistema de maneira clara e colaborativa. Esses cenários são transformados em scripts de teste automatizados que validam se o sistema funciona conforme os requisitos de negócio. O BDD facilita a comunicação entre equipes técnicas e não técnicas, garantindo que todos compreendam o comportamento esperado da aplicação, além de promover a criação de testes automatizados que verificam a funcionalidade do <i>software</i> de forma contínua e eficiente.
Pré-condição:	Histórias de usuários e critérios de aceitação especificados.
Entradas:	Cenários de testes em BDD.
Saídas:	Testes automatizados.
Ações:	Criação dos testes automatizados na linguagem do <i>Robot framework</i> .
Ferramentas:	<i>Robot Framework</i> .
Papéis Envolvidos	<ul style="list-style-type: none">● Analista de Qualidade;

Fonte: Próprio Autor.

4.2.7 Revisão das Funcionalidades Criadas

Nesta atividade, já com os testes de aceitação implementados e executados, será feita a apresentação das funcionalidades, onde o time poderá verificar através das evidências geradas pelos testes, de que a aplicação está atendendo aos critérios de aceitação e também utilizar como artefato o *log* de execução dos testes para cada funcionalidade verificada, contendo o que foi testado, como foi testado e o *status* de cada critério de aceitação, cabe ao analista de qualidade procurar deixar o *log* legível para qualquer integrante do time e *stakeholders* da aplicação, caso haja algum reajuste nos critérios de aceitação ou na própria funcionalidade, então volta para a Etapa de Preparação do *Backlog* da *Sprint*, da seção 4.2.3, estando a apresentação validada, o produto é incrementado, como mostrado na Figura 41.

Figura 41 - Revisão das Funcionalidades Criadas



Fonte: Próprio Autor.

4.2.7.1. Demonstrar a Funcionalidade

Demonstrar a funcionalidade consiste em apresentar ao time e aos *stakeholders* as funcionalidades desenvolvidas durante a *sprint*. Essa demonstração pode ser realizada por meio da execução da aplicação ou da apresentação dos resultados obtidos nos testes de aceitação automatizados.

O objetivo desta atividade é permitir que todos os envolvidos visualizem o comportamento da funcionalidade, compreendam o que foi implementado e confirmem que a solução está alinhada com as expectativas definidas no início do desenvolvimento.

4.2.7.2. Exibir os Critérios de Aceitação Atendidos

Consiste em apresentar as evidências geradas pelos testes de aceitação automatizados, demonstrando quais critérios foram validados com sucesso. Para isso, utilizam-se os logs de execução dos testes, que detalham o cenário testado, os passos executados e o resultado de cada critério de aceitação.

Cabe ao analista de qualidade garantir que esses registros estejam claros e legíveis, de modo que possam ser compreendidos por qualquer integrante do time ou *stakeholder*. Ferramentas como o *plugin TestDoc* do *Robot Framework* podem ser utilizadas para gerar relatórios em formato HTML (Linguagem de Marcação de HiperTexto), que podem ser publicados como documentação da funcionalidade, conforme exemplificado na Figura 42.

Figura 42 - Exemplo do log do framework Robot

PBI 8301 CRUD Tipo Opção Log

Generated
20240121 15:42:28 UTC-03:00

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	4	4	0	0	00:01:09	<div style="width: 100%; height: 10px; background-color: green;"></div>

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						<div style="width: 100%; height: 10px; background-color: #ccc;"></div>

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
PBI 8301 CRUD Tipo de Opção	4	4	0	0	00:01:09	<div style="width: 100%; height: 10px; background-color: green;"></div>

Test Execution Log

SUITE PBI 8301 CRUD Tipo de Opção

Full Name: PBI 8301 CRUD Tipo de Opção

Source: [C:\Testes\Bifrost_Glitnir.QualityAssurance\RobotFramework\Tests\IPBI_8301_CRUD_TipoOpcao.robot](#)

Start / End / Elapsed: 20240121 15:41:19.110 / 20240121 15:42:28.272 / 00:01:09.162

Status: 4 tests total, 4 passed, 0 failed, 0 skipped

TEST Cenário: Cadastrar tipo de opcao

Full Name: PBI 8301 CRUD TipoOpcao.Cenario: Cadastrar tipo de opcao

Start / End / Elapsed: 20240121 15:41:19.331 / 20240121 15:41:36.149 / 00:00:16.818

Status: PASS

- + **SETUP** resource .Abrir navegador
- + **KEYWORD** Bifrost_configuracoes . Dado que esteja no cadastro de tipo de opcao
- + **KEYWORD** Bifrost_configuracoes . Quando preencher os campos opcoes com dados validos e confirmar
- + **KEYWORD** Bifrost_configuracoes . Entao deve ser exibida a mensagem "Registro inserido com sucesso!"
- + **TEARDOWN** resource . Fechar navegador

+ **TEST** Cenário: Visualizar o tipo de opcao cadastrado

+ **TEST** Cenário: Alterar tipo de opcao

+ **TEST** Cenário: Excluir tipo de opcao

Fonte: Próprio Autor.

4.2.7.3. Verificar Reajuste nos Critérios de Aceitação ou Funcionalidade

Após a apresentação das funcionalidades e dos resultados dos testes, deve-se verificar a necessidade de reajustes nos critérios de aceitação ou na própria funcionalidade. Essa análise ocorre a partir do *feedback* do time e dos *stakeholders*, que podem identificar ajustes necessários para melhor atender às necessidades do negócio.

Caso sejam identificadas inconsistências, mudanças de entendimento ou novas necessidades, a funcionalidade não é considerada validada, sendo necessário realizar ajustes antes de sua aprovação final.

4.2.7.4. Voltar para Subprocesso Preparação do *Backlog* da *Sprint*

Quando há necessidade de reajuste nos critérios de aceitação ou na funcionalidade, o fluxo retorna para o subprocesso de Preparação do *Backlog* da *Sprint*, descrito na Seção 4.2.3. Nessa etapa, os itens do *backlog* são reavaliados, refinados e priorizados novamente, incorporando as mudanças identificadas durante a revisão.

Esse retorno garante que as alterações sejam tratadas de forma estruturada, mantendo a rastreabilidade entre requisitos, testes e funcionalidades.

4.2.7.5. Incrementar no Produto

Quando a funcionalidade é validada e não há necessidade de reajustes, procede-se com a atividade de incrementar no produto. Isso significa que a funcionalidade revisada e aprovada passa a fazer parte do incremento do produto ao final da *sprint*.

Esse incremento representa a entrega de valor ao negócio, assegurando que a funcionalidade incorporada atenda aos critérios de aceitação estabelecidos e esteja em conformidade com os testes executados, conforme ilustrado no fluxo da Figura 42.

Uma visão geral da Revisão das Funcionalidades Criadas é apresentada na Tabela 18.

Tabela 18 - Resumo da Revisão das Funcionalidades Criadas

REVISAR AS FUNCIONALIDADES CRIADAS	
Propósito:	Revisar as funcionalidades criadas verificando se cada uma atende corretamente aos requisitos definidos, garantindo que os cenários de testes, as regras de negócio e as interações do usuário sejam contempladas. Isso inclui testes funcionais para validar o comportamento esperado, identificação de bugs ou inconsistências, revisão da interface do usuário, validação de dados e ajustes de integração com APIs. A revisão também garante que as funcionalidades sejam consistentes, seguras e fáceis de usar, além de confirmar a

	conformidade com os critérios de aceite e o desempenho adequado em diversos cenários.
Pré-condição:	Cenários de testes em BDD e Histórias de usuário refinadas.
Entradas:	Funcionalidades e testes automatizados prontos.
Saídas:	<i>Log</i> de execução dos testes.
Ações:	Utilizar os testes de aceitação automatizados como demonstração, gerando o <i>log</i> de execução de testes como documentação.
Ferramentas:	Foi utilizado o <i>Robot Framework</i> .
Papéis Envolvidos	<ul style="list-style-type: none"> ● <i>Product Owner</i> (PO) ou Analista de Requisitos; ● Analista de Qualidade; ● Desenvolvedores.

Fonte: Próprio Autor.

5. IMPLANTAÇÃO DO PROCESSO IAP

Seguindo as etapas da pesquisa-ação apresentadas na Seção 1.3, inicialmente foi realizada a investigação do processo de desenvolvimento da empresa, com o intuito de analisar as práticas adotadas, bem como identificar desafios e limitações, conforme descrito no Capítulo 3. A partir dessa análise, aliada às lacunas identificadas na literatura sobre especificação de requisitos em projetos ágeis, foi proposto o processo IAP, apresentado no Capítulo 4.

O presente capítulo tem como objetivo descrever a implantação do processo IAP na empresa, evidenciando sua aplicação no ambiente real de desenvolvimento. Para isso, são abordadas as etapas de preparação, a capacitação da equipe, o acompanhamento da utilização do processo ao longo das sprints e as dificuldades encontradas durante sua adoção.

5.1. Contexto da implantação

A implantação do processo na empresa teve a participação ativa da pesquisadora, fazendo intervenções, sempre que necessário para esclarecer dúvidas, orientar e adequar às práticas propostas às necessidades da empresa, sendo um aspecto característico de uma pesquisa-ação.

A implantação do processo de Integração do ATDD e PBB foi realizada de forma integrada à metodologia *Scrum* que já era adotada pela empresa para promover ciclos de desenvolvimento iterativos e incrementais, favorecendo a colaboração e a adaptação contínua. Para o gerenciamento das atividades dos stakeholders envolvidos na implantação, foi utilizada a ferramenta *Azure DevOps* que também já era adotada pela empresa.

O estudo foi realizado com um grupo específico de 7 pessoas envolvendo PO, Analistas de Qualidade, Desenvolvedores e um Gerente de Projetos. O acompanhamento da utilização do processo na empresa se deu durante 3 (três) sprints de 15 dias cada e foi baseada em boas práticas de times ágeis e na necessidade de equilibrar aprendizado, aplicação e ajustes sem comprometer a produtividade da equipe, abrangendo as etapas de capacitação, aplicação do processo e monitoramento contínuo. A escolha desse período também se alinha à cadência natural de equipes que utilizam *Scrum*, onde *sprints* de duas semanas são comuns para permitir entregas frequentes e revisões contínuas.

5.2. Capacitação no Processo

A capacitação dos envolvidos teve por objetivo possibilitar que todos os membros da equipe, independentemente de seu papel, tivessem a compreensão dos fundamentos teóricos e práticos do Processo proposto. A capacitação não apenas promoveu o alinhamento conceitual entre os participantes, mas também estabeleceu uma base sólida para a aplicação eficaz das práticas no dia a dia do desenvolvimento.

Dado o caráter colaborativo do IAP, a capacitação foi fundamental para que o time conseguisse integrar as práticas ATDD e PBB de forma fluida. A prática PBB, por exemplo, exige que os membros saibam construir e priorizar um *backlog* de forma colaborativa, enquanto o ATDD depende de um entendimento claro sobre a criação e a validação de critérios de aceitação e cenários de teste. Assim, é indispensável que todos compreendam o papel que desempenham nesse processo integrado.

O treinamento foi conduzido por meio de videoconferência, em sessões conduzidas ao longo de dois dias, sendo organizada da seguinte maneira:

- No primeiro dia, o treinamento focou na construção do *backlog* do produto, priorização do *backlog* e preparação do *backlog da sprint*;
- No segundo dia, o treinamento focou no refinamento das histórias de usuário para o BDD, o desenvolvimento de sistema e testes automatizados e por fim, a revisão das funcionalidades criadas.

Um total de 7 pessoas participaram do treinamento, abrangendo desenvolvedores, gerente de projeto, analistas de requisitos e QA. Esta diversidade de participantes contribuiu para facilitar a compreensão do novo conjunto de diretrizes para toda a equipe envolvida.

Além de abordar aspectos técnicos, a capacitação também teve como objetivo fomentar uma cultura de colaboração e aprendizado contínuo. Durante essa etapa, a troca de experiências entre os membros da equipe contribuiu para a criação de um ambiente propício à colaboração de melhorias e à resolução de problemas. Por meio de sessões teóricas, práticas e acompanhamento constante, a capacitação buscou potencializar que o time estivesse preparado para os desafios de aplicar o processo em um ambiente real.

5.2.1. Estratégia adotada para a Capacitação

O modelo de capacitação foi estruturado sob um planejamento pedagógico que visou potencializar a compreensão dos membros do time sobre os fundamentos do processo IAP. A estratégia foi dividida em momentos teóricos e práticos, respeitando a divisão cronológica de dois dias estabelecida anteriormente.

No primeiro dia, o foco recaiu sobre a Construção e Priorização do *Backlog*. Inicialmente, foram realizadas sessões teóricas para apresentar as práticas PBB e ATDD, explorando suas interseções e o impacto esperado no fluxo de trabalho. Na sequência, conduziram-se oficinas práticas utilizando o PBB Canvas, onde o time, de forma colaborativa, exercitou a criação de histórias de usuário detalhadas e a simulação de sessões de priorização para facilitar o entendimento da sequência lógica do processo.

No segundo dia, as atividades concentraram-se no Refinamento para BDD e Ciclo de Desenvolvimento. Foram detalhados os procedimentos para a criação de cenários de teste e a integração

com o desenvolvimento automatizado. Durante todo o treinamento, manteve-se um acompanhamento contínuo, com sessões de perguntas e respostas para revisar o progresso e sanar dúvidas em tempo real. Como resultado desse acompanhamento, identificou-se a necessidade de adaptações nos *templates* originais para assegurar a conformidade com o ambiente operacional da empresa. Essas modificações, que incluíram o ajuste de campos no *Azure DevOps* para suportar nativamente os Critérios de Aceite e os cenários BDD, foram fundamentais para que a estratégia de capacitação contribuísse para o engajamento e a confiança da equipe na utilização do processo.

Após a capacitação, antes de iniciar na semana seguinte a primeira *sprint*, foram realizadas adaptações nos *templates* versão Word para o PBI do *Azure DevOps*. Os referidos *templates* estão previstos nos subprocessos “PREPARAÇÃO DO BACKLOG DA SPRINT” e “REFINAMENTO DAS HISTÓRIAS DE USUÁRIO PARA O BDD”, descritos previamente nas seções 4.2.3 e 4.2.4, respectivamente, e que estão disponibilizados nos Apêndices B e C. O *template* disponível no Apêndice B da História do Usuário foi adaptado para dentro do PBI do *Azure DevOps* através dos campos “*Description*” e “*Acceptance Criteria*”. No “*Description*” foi escrito a história de usuário e colocado o *link* do protótipo da interface (se houvesse) e o campo “*Acceptance Criteria*” foi escrito os critérios de aceitação da funcionalidade, como mostrado na Figura 43.

Figura 43 – Campos utilizados para o Apêndice B no *Azure DevOps*

9328 Visualização de Faturas no Portal do Cliente

No one selected 0 Comments Add Tag

State ● New Area GDS\QA\QA-Produto

Reason Moved to the backlog Iteration GDS\Project Sprints\2025-02-03

Description

Descrição da História de Usuário:
Como um cliente autenticado, eu quero visualizar minhas faturas mensais no portal, para poder acompanhar os valores cobrados e efetuar os pagamentos no prazo.

Acceptance Criteria

Critérios de Aceite:

1. O cliente autenticado deve poder acessar a seção "Minhas Faturas" no menu principal.
2. A lista de faturas deve ser apresentada em ordem cronológica, da mais recente para a mais antiga.
3. Cada fatura deve exibir informações como: mês de referência, valor total, status (paga/em aberto/vencida) e data de vencimento.
4. Deve haver um botão "Visualizar PDF" para cada fatura, permitindo o download da versão completa.

Fonte: Próprio Autor.

Na Figura 44, é apresentado mais um campo, chamado de “BDD”, onde foi preenchido de acordo com o *template* do Apêndice C dos cenários de testes.

Figura 44 – Campo utilizado para o Apêndice C no Azure Devops

9328 Visualização de Faturas no Portal do Cliente

No one selected 0 Comments Add Tag

State New Area GDS\QA\QA-Produto
Reason Moved to the backlog Iteration GDS\Project Sprints\2025-02-03

Critérios de Aceite:

1. O cliente autenticado deve poder acessar a seção "Minhas Faturas" no menu principal.
2. A lista de faturas deve ser apresentada em ordem cronológica, da mais recente para a mais antiga.
3. Cada fatura deve exibir informações como: mês de referência, valor total, status (paga/em aberto/vencida) e data de vencimento.
4. Deve haver um botão "Visualizar PDF" para cada fatura, permitindo o download da versão completa.

BDD ↗

BDD – Cenários de Teste (QA):

Cenário 1: Acesso à seção "Minhas Faturas"
Dado que o cliente está logado no portal
E está na página inicial do sistema
Quando acessar o menu principal e clicar na opção "Minhas Faturas"
Então o sistema deve redirecioná-lo para a página de listagem de faturas

Cenário 2: Exibição da lista de faturas em ordem cronológica
Dado que o cliente está na página "Minhas Faturas"
E existem múltiplas faturas cadastradas em seu histórico
Quando a lista for carregada
Então as faturas devem ser exibidas da mais recente para a mais antiga

Cenário 3: Exibição de informações detalhadas por fatura
Dado que o cliente está na listagem de faturas
Quando visualizar a fatura referente a um mês específico
Então o sistema deve mostrar o mês de referência, valor total, status e data de vencimento dessa fatura

Fonte: Próprio Autor.

Para ter acesso aos *templates*, o time acessava uma pasta disponibilizada no *Google Drive*⁸, onde era possível baixar o *template* que seria usado para cada atividade específica.

5.2.2. Dificuldades e Lições Aprendidas Durante a Capacitação

A capacitação, realizada em sessões conduzidas em dois dias, apresentou desafios significativos relacionados à diversidade de perfis dos participantes, que possuíam níveis distintos de familiaridade com as práticas ATDD e PBB. Foi observado que conceitos-chave, especialmente no que tange à

⁸ <https://drive.google.com>

integração entre as práticas, geraram dúvidas que exigiram explicações mais interativas e a utilização de exemplos práticos. Essa situação evidenciou a importância de uma abordagem de ensino adaptável para atender aos diferentes níveis de aprendizagem dos integrantes do time.

Outro desafio relevante enfrentado durante a etapa de capacitação foi a gestão do tempo nas sessões de treinamento. Apesar de um cronograma detalhadamente estruturado, atividades práticas como a construção do PBB Canvas consumiram mais tempo do que o inicialmente planejado, devido ao elevado volume de perguntas e à intensa participação da equipe. Esse engajamento, embora positivo, exigiu ajustes pontuais na programação, resultando na redução do tempo destinado a outras atividades, como a simulação de refinamento do *backlog*. Considerou-se a possibilidade de expandir o tempo total da capacitação, mas tal ajuste não foi viável naquele momento devido à limitação de agenda dos participantes e à necessidade de manter a cadência das *sprints* planejadas.

Para evitar prejuízos à qualidade do aprendizado, foram adotadas estratégias compensatórias, como o reforço de conteúdos-chave em momentos assíncronos e a disponibilização de materiais de apoio complementares. Dessa forma, buscou-se garantir que os objetivos centrais da capacitação fossem plenamente atendidos, mesmo diante dos ajustes necessários no tempo de execução.

Entre as lições aprendidas, destacou-se a eficácia de incorporar momentos regulares de pausa para *feedback*, permitindo ajustes imediatos no andamento do treinamento. Além disso, o uso de exemplos práticos relacionados ao contexto real da equipe mostrou-se essencial para facilitar o engajamento e o entendimento dos conceitos. Assim, foi possível perceber que, além da transmissão de conhecimento técnico, um treinamento bem-sucedido requer a habilidade de fomentar a colaboração, promover um ambiente participativo e adaptar-se às dinâmicas do grupo.

5.3. Monitoramento da Utilização do Processo

O monitoramento da utilização do processo IAP foi essencial para acompanhar sua aplicação prática e identificar oportunidades de ajuste ao longo do ciclo de desenvolvimento. Durante as três *sprints*, foram utilizadas métricas de acompanhamento constantes, tais como: quantidade de histórias de usuário refinadas, criação e validação de critérios de aceitação, quantidade de cenários BDD elaborados, cobertura de testes automatizados e tempo médio entre o refinamento e a validação de cada história. A manutenção dessas métricas em todas as *sprints* permitiu uma avaliação comparativa do progresso da equipe, facilitando o diagnóstico de gargalos e o entendimento de como o processo estava sendo assimilado e aplicado.

Ao todo, foram elaborados 24 PBIs, correspondendo a 24 histórias de usuário, acompanhadas por 110 critérios de aceitação e 110 cenários em BDD, resultando numa média de aproximadamente 4,6 critérios por história. Esses números evidenciam a aplicação sistemática do processo IAP durante as *sprints*, especialmente no que se refere à definição estruturada de critérios de aceitação e à construção de cenários em BDD para cada história de usuário.

Diferentemente do processo anteriormente utilizado no qual não havia padronização na definição de critérios de aceitação nem na elaboração de cenários de teste, os dados apresentados demonstram que, com o IAP, todas as histórias passaram a conter critérios e cenários associados, indicando maior consistência na especificação dos requisitos. Ainda que não haja métricas quantitativas equivalentes do processo anterior para comparação direta, a padronização observada nesta etapa representa uma evolução em relação à prática anterior, baseada predominantemente em descrições textuais informais.

A consistência na criação de critérios e cenários em todas as histórias reforça o compromisso da equipe em manter o alinhamento proposto pelo processo, ainda que, em alguns momentos, tenham sido necessários ajustes na forma de escrita e na definição de granularidade das histórias.

Reuniões de acompanhamento foram realizadas ao final de cada *sprint* para discutir os desafios enfrentados e ajustar o processo conforme necessário. Essas reuniões incluíram demonstrações de funcionalidades desenvolvidas e análises de *logs* de testes automatizados para validar se os critérios de aceitação estavam sendo atendidos. A gerente de projeto desempenhou um papel fundamental na consolidação dessas informações e no alinhamento das prioridades para as próximas *sprints*.

Um dos resultados positivos observados foi o aumento da clareza nos requisitos, que reduziu significativamente as ambiguidades durante o desenvolvimento. Além disso, a automatização dos testes de aceitação proporcionou um feedback mais rápido sobre a qualidade das entregas, permitindo correções imediatas quando necessário.

Após o encerramento do período de monitoramento da utilização do processo, foi dado início à segunda fase de coleta de dados, sendo esta baseada em novas entrevistas semiestruturadas para coletar a visão do time em relação às melhores práticas, dificuldades e limitações da abordagem proposta. Ao todo, participaram desta etapa 7 profissionais, incluindo *Product Owner*, QA, desenvolvedores e gerente de projetos.

Cada entrevista teve duração média de aproximadamente 1 (uma) hora, sendo necessárias duas jornadas para sua realização, tendo em vista a disponibilidade da equipe. Todas as entrevistas foram conduzidas por videochamadas, utilizando a ferramenta *Google Meet*, e posteriormente transcritas.

A análise das entrevistas permitiu identificar padrões nas percepções dos participantes. De forma geral, observou-se predominância de avaliações positivas em relação ao processo IAP, especialmente quanto à clareza dos requisitos e ao alinhamento entre os membros da equipe. Por outro lado, as principais dificuldades relatadas estiveram associadas à adaptação inicial ao processo e à necessidade de maior padronização na escrita dos critérios de aceitação. Esses resultados foram consolidados por meio de categorização das respostas e são apresentados de forma detalhada nas seções seguintes, incluindo a quantificação das percepções dos participantes.

Por fim, o monitoramento demonstrou que foi fundamental para captar percepções e promover ajustes pontuais. As entrevistas reforçaram a importância de se manter um ciclo contínuo de *feedback* e aprendizagem, característica central dos métodos ágeis, permitindo a evolução do processo de forma colaborativa e eficaz.

5.4. Dificuldades Identificadas na Implantação

Durante a implantação, algumas dificuldades foram identificadas pela pesquisadora. A principal foi a resistência inicial de parte dos membros da equipe em adotar o processo, especialmente devido à curva de aprendizado associada às práticas ATDD e PBB. Essa resistência foi observada em 3 dos 7 participantes, predominantemente entre desenvolvedores com menor experiência prévia em testes de aceitação e práticas ágeis estruturadas, além de um profissional em nível pleno que não possuía familiaridade com BDD.

Essa dificuldade esteve relacionada, principalmente, à necessidade de mudança na forma de especificar requisitos e na adaptação ao uso de critérios de aceitação e cenários estruturados. No entanto, a resistência foi gradualmente superada por meio de treinamentos pontuais e acompanhamento individualizado, o que contribuiu para a evolução do entendimento e da adoção do processo ao longo das *sprints*.

Outro desafio enfrentado durante a aplicação do processo IAP foi a necessidade de ter um nível de detalhamento adicional, especialmente na fase de refinamento do *backlog* e na definição dos critérios de aceitação. Isso demandou um esforço adicional para a equipe, mas que é necessário para se ter a integração entre as equipes de desenvolvimento e testes.

As reuniões de refinamento, por exemplo, precisaram ser reformuladas para incluir etapas específicas de construção e validação dos critérios de aceitação baseados no ATDD, o que aumentou o tempo necessário para cada sessão. Isso exigiu não apenas um novo planejamento do tempo das cerimônias, mas também mudanças na dinâmica de participação, ampliando o papel dos testadores e *stakeholders* nessas reuniões, algo que nem sempre ocorria nas práticas originais do time.

Além disso, foi necessário um esforço adicional de coordenação entre os *Product Owners* (POs) e a gerente de projeto, para garantir que os novos artefatos gerados – como os resultados do PBB Canvas e os critérios ATDD – estivessem devidamente integrados ao *backlog* oficial e sincronizados com o planejamento das *sprints*. Isso incluiu adaptações nos *templates* de histórias, definição de responsáveis por etapas intermediárias e alinhamento frequente entre os envolvidos para evitar retrabalho.

A limitação de tempo foi gerenciada por meio de um conjunto de estratégias coordenadas entre a equipe técnica e a gestão do projeto, com foco em minimizar o impacto sobre as entregas regulares e garantir a adesão ao novo processo. As principais ações adotadas foram:

- Priorização e alinhamento com a liderança: A gerente de projeto atuou de forma ativa junto aos *Product Owners* e *stakeholders* para redefinir prioridades, estabelecendo o que era essencial para cada sprint e quais atividades poderiam ser postergadas. Isso permitiu que a equipe tivesse espaço para participar das atividades do IAP sem comprometer as metas críticas do projeto.
- Ajuste da carga de trabalho: As *sprints* seguintes à introdução do IAP foram planejadas com capacidade reduzida, reconhecendo o tempo necessário para capacitações, reuniões de refinamento detalhado e adaptação. Houve uma distribuição mais equilibrada das tarefas entre os membros do time, evitando a sobrecarga de indivíduos específicos.
- Alocação de tempo dedicado: Foram criadas janelas fixas na agenda da equipe para as atividades do processo, como as sessões de PBB Canvas e elaboração dos critérios de aceitação com base no ATDD. Isso ajudou a dar previsibilidade e organização, evitando conflitos com outras responsabilidades.
- Capacitações integradas à rotina: Em vez de treinamentos longos e isolados, a equipe adotou um formato de capacitação contínua e aplicada. O conhecimento sobre o processo era transmitido e praticado diretamente durante as atividades reais do projeto, promovendo aprendizado no contexto do trabalho.
- Monitoramento contínuo e feedback rápido: Durante os primeiros ciclos, a equipe manteve feedbacks frequentes sobre a gestão do tempo, permitindo ajustes ágeis na organização das tarefas e nas cerimônias Scrum. Isso possibilitou adaptações rápidas sempre que surgiam conflitos ou atrasos.

Essas estratégias foram fundamentais para equilibrar a adoção do IAP com a rotina de trabalho da equipe, transformando um desafio significativo em uma oportunidade de amadurecimento organizacional e melhoria contínua.

Durante o período de monitoramento da utilização do processo, entre a segunda e a terceira *sprints*, foi possível observar diferenças importantes, especialmente no nível de maturidade da equipe na aplicação dos conceitos da abordagem. Essa evolução foi evidenciada por métricas de acompanhamento definidas no estudo, como: quantidade de histórias de usuário refinadas com critérios de aceitação completos, número de cenários BDD elaborados por história, redução no tempo médio entre o refinamento e a validação das histórias e maior estabilidade na escrita dos critérios de aceitação.

Na terceira *sprint*, observou-se maior consistência na definição dos critérios e na construção dos cenários, além de uma redução na necessidade de retrabalho durante a validação, indicando maior compreensão do processo. O time também demonstrou maior agilidade na organização do *backlog*, refletida na menor necessidade de ajustes estruturais durante o planejamento da sprint. Esses fatores evidenciam um progresso gradual da equipe, decorrente da aprendizagem acumulada ao longo das *sprints* anteriores.

As melhorias observadas incluíram a redução de retrabalho, a elaboração mais precisa dos critérios de aceitação e uma colaboração mais fluida entre os papéis envolvidos no desenvolvimento. Esses avanços ocorreram principalmente porque os membros da equipe se sentiram mais confiantes para aplicar o modelo proposto, além de estarem mais alinhados quanto ao propósito e funcionamento do processo.

A evolução entre as *sprints* também se manifestou na forma como as cerimônias do *Scrum* foram adaptadas: o refinamento passou a ser mais objetivo, e o time conseguiu antecipar dúvidas e dependências, otimizando o planejamento das tarefas. Esse aspecto técnico foi fundamental para melhorar o fluxo de trabalho e manter a cadência das entregas.

A ferramenta *Azure DevOps* teve um papel crucial nesse processo de amadurecimento técnico. Sua utilização permitiu centralizar as informações, acompanhar o andamento das atividades, documentar as histórias de usuários e os critérios de aceitação e facilitar a rastreabilidade das mudanças. Com isso, a equipe pôde monitorar com mais precisão a evolução das histórias de usuário e identificar gargalos ao longo da *sprint*.

Em resumo, a terceira *sprint* refletiu um nível maior de integração do processo IAP na rotina da equipe, resultando em entregas mais bem alinhadas aos requisitos definidos e fortalecendo a colaboração e a transparência no time.

As experiências, desafios e resultados obtidos durante a implantação da abordagem, conforme apresentado neste capítulo, fornecem subsídios importantes para futuras pesquisas que tenham como objetivo a implementação do processo IAP.

6. AVALIAÇÃO DO PROCESSO IAP

Este capítulo apresenta a avaliação do processo de integração das práticas ATDD e PBB, com base no protocolo de pesquisa-ação e tem por objetivo responder à questão definida anteriormente na seção 1.2.2: “Como as práticas ATDD e PBB podem ser utilizadas de forma integrada e sistemática como estratégia para especificar requisitos em projetos ágeis, atendendo a equipes de desenvolvimento e qualidade?”. Com o objetivo de compreender seu impacto no desenvolvimento ágil, a avaliação do processo se concentra nos resultados obtidos, comparações com as práticas antes da implantação do IAP, avaliação de critérios de pesquisa-ação e reflexões sobre ameaças à validade.

6.1. Coleta e Análise dos dados

O processo IAP se propõe a contribuir na clareza dos requisitos e na automação dos critérios de aceitação, alinhando os esforços do time de desenvolvimento e testes. Para validar o processo proposto, foi feita sua implantação numa empresa brasileira de *software* que atua na automação de transações para o comércio exterior com o objetivo de investigar como um time de desenvolvimento, composto por PO, Analistas de Qualidade, Desenvolvedores e um Gerente de Projetos, lida com a especificação de requisitos e se já utilizava ATDD ou PBB em projetos ágeis.

O trabalho de Medeiros (2017) foi utilizado como referência nesta pesquisa no que diz respeito aos instrumentos para coleta e análise dos dados. Foi adotada a categorização em dois fatores: "Conteúdo", que se refere à forma como os requisitos são descritos, e "Estrutura", que trata do tipo de informação presente e da organização dos elementos na especificação. Essa abordagem metodológica clara e objetiva possibilitou uma avaliação mais sistemática e fundamentada dos impactos da implantação do processo IAP (Integração entre ATDD e PBB) no contexto ágil. A estrutura proposta por Medeiros (2017) mostrou-se adequada para capturar nuances da percepção da equipe sobre os requisitos, oferecendo uma base teórica consistente e compatível com os objetivos desta dissertação.

O estudo de Medeiros et al. (2018) também foi adotado como base para a definição dos indicadores a serem avaliados por apresentar uma análise empírica sólida e validada a partir de casos reais do contexto industrial, o que o torna relevante para aplicações práticas, como a desenvolvida nesta pesquisa. A partir da análise cruzada de seis estudos de caso industriais, os autores identificaram fatores de qualidade amplamente reconhecidos na literatura e aplicáveis à especificação de requisitos de *software*, entre eles, Completude, Legibilidade, Objetividade, Conformidade e Volatilidade:

- **Completude:** Foi utilizado para medir se os PBI's no modelo IAP, fornecem todas as informações necessárias para que o time possa progredir no desenvolvimento sem necessidade de informações adicionais. Isso inclui detalhes sobre funcionalidades, comportamentos esperados e dependências.

- **Conformidade:** Avalia um alinhamento entre o que foi produzido e o que é esperado operacionalmente dentro do contexto do projeto. No caso desta pesquisa, isso foi potencializado pela utilização de *templates* padronizados no *Azure DevOps*, pelo alinhamento constante entre os membros da equipe sobre o formato e o nível de detalhamento desejado nos critérios de aceitação, e pela validação conjunta dos artefatos durante as reuniões de refinamento. Assim, a conformidade foi observada sempre que as histórias de usuário, critérios e cenários estavam não apenas tecnicamente corretos, mas também adequados à linguagem, estrutura e utilidade prática adotadas pelo time no seu fluxo de trabalho.
- **Legibilidade:** Este indicador foi utilizado para avaliar quão facilmente o time entende e interpreta os PBI's gerados no modelo IAP. A alta legibilidade é alcançada através de uma apresentação clara das informações, organização lógica e uso de uma linguagem acessível, permitindo que o time compreenda rapidamente os requisitos.
- **Objetividade:** Refere-se à clareza e precisão com que os requisitos são definidos. Um IAP 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.
- **Volatilidade:** Indica a frequência e o volume de alterações que um PBI 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 requisitos para atender às expectativas dos *stakeholders*.

Esses fatores foram escolhidos por sua relevância na avaliação do conteúdo das especificações e por estarem diretamente alinhados aos objetivos deste estudo, que buscou analisar a qualidade dos documentos produzidos com a aplicação do processo IAP no ambiente do *Azure DevOps*.

6.1.1. Conteúdo

Durante a aplicação da abordagem, o time realizou uma avaliação detalhada de todos os 24 documentos de especificação produzidos dentro do *Azure DevOps* (Histórias de Usuários, Critérios de aceitação e Cenários em BDD).

A avaliação do conteúdo das especificação de requisitos produzida utilizando o processo IAP foi realizada pelo time por meio de entrevistas semiestruturadas, conforme apresentado no Apêndice D.

As perguntas do questionário foram desenhadas para avaliar diretamente cada um dos indicadores, utilizando tanto escalas quantitativas e qualitativa juntas, como a de 1 (inadequado) a 5 (muito adequado), e respondendo o porquê da escolha, a exemplo da Q9: “Numa escala de um

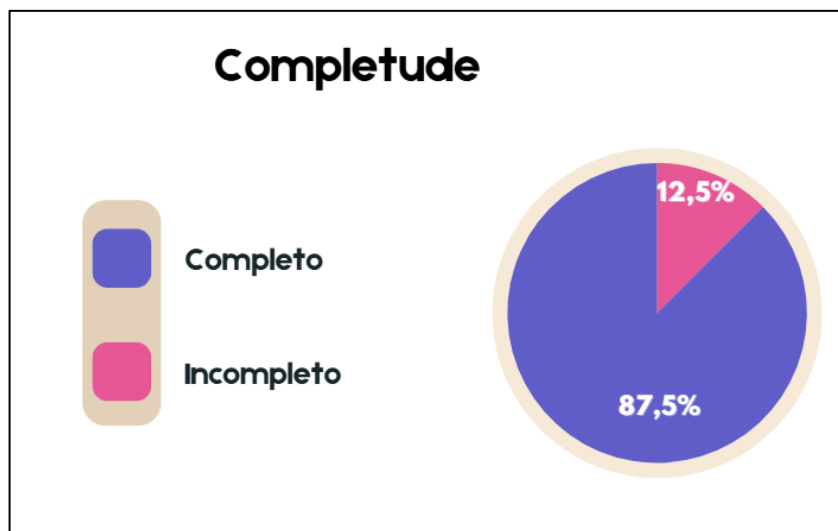
(inadequado) a cinco (muito adequado), como avalia a estrutura da integração das práticas ATDD e PBB? Porquê?”.

Já no que tange as questões abertas, as perguntas permitiram uma análise mais qualitativa, a exemplo da Q8: “Como você avalia o ATDD estruturado por critérios de aceitação, cenários de testes e testes automatizados?” e da Q11: “Quais as diferenças do processo para integração das práticas ATDD e PBB em relação ao que você utilizava anteriormente?”.

Note, que a objetividade e a legibilidade foram medidas através de perguntas que pediam ao time para avaliar a estrutura da integração das práticas ATDD e PBB (IAP) e sua conformidade com critérios de qualidade específicos. A escala de 1 a 5 permitiu uma avaliação padronizada, enquanto as questões abertas possibilitaram aos participantes expressarem suas percepções de maneira mais abrangente.

Para a completude, 87,5% dos requisitos desenvolvidos no processo IAP, correspondentes a 21 artefatos, foram avaliados como suficientemente detalhados para permitir o desenvolvimento sem consultas adicionais, como mostra a Figura 45. A completude de três PBI's 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 45 - Percentuais avaliados no critério de completude

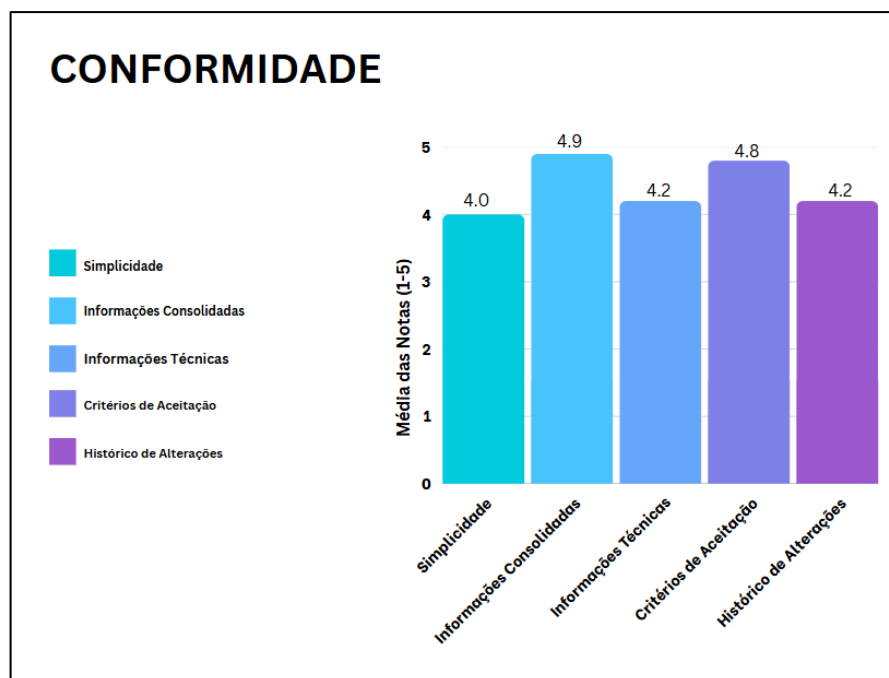


Fonte: Próprio Autor.

Na Figura 46, com base nas entrevistas do questionário, foram consolidadas as respostas referentes a Q10: “Usando uma escala de um (inadequado) a cinco (muito adequado), como você avalia a conformidade do processo para integração das práticas ATDD e PBB com cada fator de qualidade?”

(Fatores: Simplicidade, Informações Consolidadas, Informações técnicas, Critérios de Aceitação e Histórico de alterações)”.

Figura 46 – Conformidade do Processo IAP por Fator de Qualidade



Fonte: Próprio Autor.

As informações apresentadas relacionadas à conformidade foram organizadas com base na média das notas atribuídas pelos entrevistados, permitindo uma visualização geral dos resultados obtidos. Embora todos os fatores de qualidade avaliados na questão Q10 pudessem, teoricamente, alcançar a nota máxima (5), a variação nas avaliações pode ser explicada por diferentes níveis de percepção quanto à maturidade do processo em aspectos específicos.

Essa variabilidade também está diretamente relacionada ao perfil dos participantes, conforme caracterizado anteriormente, considerando fatores como nível de experiência (júnior, pleno e sênior), tempo de atuação na área e papel desempenhado no projeto. Profissionais com maior experiência ou maior proximidade com atividades de requisitos, como QA e *Product Owner*, tendem a apresentar uma visão mais crítica e detalhada do processo, enquanto membros com menor experiência ou atuação mais técnica podem avaliar positivamente aspectos relacionados à execução, mas com menor foco em questões de especificação e alinhamento de requisitos.

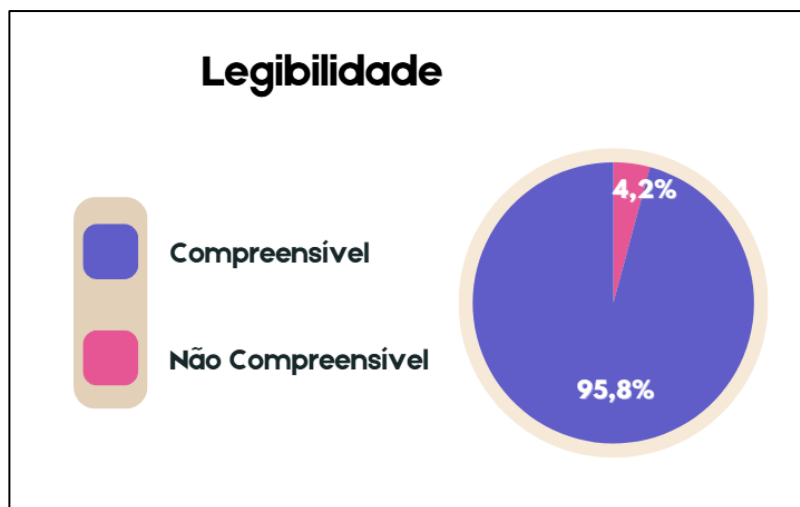
Além disso, fatores como o grau de familiaridade com o modelo IAP, a adaptação individual à nova forma de documentar e refinar requisitos e a fase de transição ainda em curso durante a coleta das respostas também podem ter influenciado essas avaliações. Aspectos como pequenas inconsistências

na consolidação de informações ou a ausência de padronização inicial em alguns critérios de aceitação podem ter gerado interpretações mais conservadoras por parte dos entrevistados.

Dessa forma, a variação observada nas notas não necessariamente indica falhas no processo, mas reflete diferentes perspectivas da equipe frente a um processo em evolução, evidenciando oportunidades de melhoria contínua, características de um ambiente ágil em fase de implantação e amadurecimento.

Em termos de legibilidade, 95,8% dos requisitos desenvolvidos no processo IAP, correspondentes a 23 artefatos, foram considerados facilmente compreensíveis, como mostra a Figura 47, indicando uma apresentação clara das informações constantes nos artefatos. No entanto, a legibilidade de um PBI foi afetada negativamente por conter mais de um requisito em um único critério de aceitação, complicando a interpretação por parte de um dos integrantes do time.

Figura 47 - Percentuais avaliados no critério de legibilidade



Fonte: Próprio Autor.

Quanto à objetividade, a análise revelou que 91,6% dos requisitos desenvolvidos no processo IAP, correspondentes a 22 artefatos, atenderam ao critério de objetividade, como mostra a Figura 48, fornecendo informações claras e diretas necessárias para a implementação. Contudo, dois PBI's foram marcados como não objetivos, devido à inclusão de informações redundantes que não agregam valor ao time.

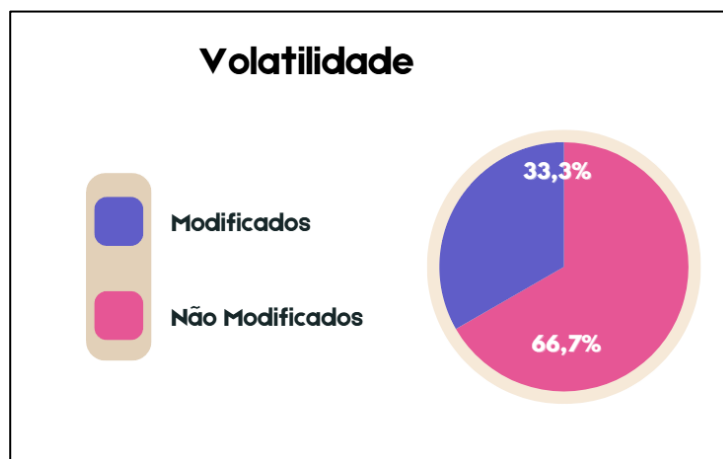
Figura 48 - Percentuais avaliados no critério de objetividade



Fonte: Próprio Autor.

A análise da volatilidade, entendida como o indicador que mede se os requisitos sofreram alterações ao longo do processo, seja para correção ou aprimoramento, identificou que 8 dos 24 artefatos avaliados foram modificados, representando 33,3% de incidência de mudanças, conforme ilustrado na Figura 49. Importa destacar que, para fins desta análise, considerou-se apenas a ocorrência ou não de modificação em cada artefato individualmente, independentemente da quantidade de alterações realizadas em um mesmo item. Ou seja, mesmo que um artefato tenha sofrido mais de uma modificação, ele foi contabilizado uma única vez como “modificado”. Essa abordagem binária (“sim” ou “não” para modificação) tem o objetivo de avaliar a estabilidade dos artefatos a partir da sua versão inicial. O resultado sugere que, embora a maioria dos PBI’s tenha sido considerada adequada desde sua concepção, uma parte significativa passou por ajustes. Isso reforça tanto o aprendizado contínuo da equipe na elicitação e documentação de requisitos quanto sua capacidade de adaptação para garantir a qualidade e alinhamento dos PBI’s com as reais necessidades do projeto e dos *stakeholders*.

Figura 49 - Percentuais avaliados no critério de volatilidade



Fonte: Próprio Autor.

6.1.2. Estrutura

O time foi entrevistado quanto à avaliação da estrutura da integração das práticas ATDD e PBB com base em todas as etapas do IAP e seus respectivos conceitos, sendo solicitado que atribuíssem uma nota em uma escala de 1 (Inadequado) a 5 (Muito Adequado), acompanhada de uma justificativa para a avaliação atribuída. De forma geral, todos os entrevistados classificaram a estrutura do IAP como “Muito Adequada”, destacando a clareza e a coerência do modelo como pontos fortes. Além da nota, os participantes relataram percepções que reforçam esse resultado positivo. Um dos entrevistados afirmou que *“a estrutura ajudou a entender o fluxo desde a concepção da ideia até o desenvolvimento, com menos retrabalho”*. Outro destacou que *“o uso do PBB Canvas facilitou muito a visualização do backlog e alinhou todo mundo sobre o que precisava ser feito”*. Esses relatos evidenciam que o modelo contribuiu para a melhoria da comunicação interna e da organização das demandas.

No que diz respeito ao ATDD estruturado por critérios de aceitação, cenários de testes e testes automatizados, e ao PBB organizado através do Canvas com os conceitos de *Product Name, Problems, Expectations, Personas, Features* e PBI, os entrevistados também reconheceram que esses elementos proporcionaram uma estrutura lógica e bem encadeada. Um dos membros da equipe comentou: *“Antes era difícil saber exatamente o que o cliente queria. Agora, com os critérios de aceitação e o Canvas, tudo ficou mais claro”*.

Essas justificativas não apenas reforçam a nota máxima atribuída, como também demonstram que a estrutura do processo IAP teve aplicabilidade prática, gerando impacto positivo no dia a dia do time e promovendo uma experiência mais eficiente de desenvolvimento ágil.

Por fim, a estrutura do IAP foi comparada com outras abordagens previamente utilizadas pelo time, com o objetivo de avaliar sua efetividade em relação aos métodos anteriormente adotados. Todos os entrevistados afirmaram que o processo proposto pelo IAP é mais apropriado, principalmente por proporcionar maior clareza, organização e rastreabilidade desde as etapas iniciais. Embora tenham reconhecido que o modelo exige um esforço inicial maior, especialmente no alinhamento entre os papéis e na definição dos critérios de aceitação, também destacaram que esse esforço é compensado por uma significativa redução de retrabalho ao longo do ciclo de desenvolvimento.

Essa percepção é reforçada por comentários como: *“No começo dá mais trabalho montar tudo, mas depois a gente percebe que não precisa voltar toda hora para corrigir ou reescrever história”*. Outro participante afirmou: *“Com o IAP, os requisitos já vêm mais redondos. Antes a gente discutia muito durante o desenvolvimento porque as histórias estavam incompletas”*. Houve ainda quem destacasse a melhoria no entendimento coletivo das tarefas: *“Ficou muito mais fácil alinhar todo mundo desde o início, coisa que não acontecia com os métodos que usávamos antes”*.

Esses relatos demonstram que, mesmo exigindo mais dedicação no início do processo, o IAP oferece benefícios práticos e consistentes a médio e longo prazo, sendo percebido pela equipe como uma abordagem superior às anteriores em termos de estrutura, clareza e produtividade.

Os resultados das entrevistas indicaram que o IAP atendeu às expectativas da equipe no que se refere à sua estrutura. Os participantes relataram que o modelo proposto foi eficaz em promover alinhamento, organização e clareza durante o desenvolvimento. Um dos aspectos mais destacados foi o uso de critérios de aceitação bem definidos em conjunto com o PBB Canvas, o que resultou em maior precisão nos testes e melhor direcionamento das entregas.

Esse alinhamento entre teoria e prática é evidenciado pelos relatos dos próprios entrevistados. Um dos membros da equipe comentou: *“O uso de critérios de aceitação claros e o suporte do PBB Canvas aumentaram a precisão dos testes”*, demonstrando como a estrutura contribuiu diretamente para a qualidade técnica das validações. Outro reforçou a melhoria no refinamento e gestão do *backlog*: *“A colaboração durante o refinamento do backlog e os critérios claros de aceitação melhoraram a clareza e a gestão das prioridades”*.

Além disso, foi destacado o impacto positivo na comunicação e na redução de incertezas entre os membros da equipe. Um participante afirmou: *“Os critérios de aceitação claros e o alinhamento promovido pelo PBB Canvas facilitaram a comunicação e reduziram incertezas”*. Esses depoimentos revelam que a estrutura do IAP não apenas atendeu, mas superou as expectativas iniciais, ao tornar o processo de especificação e desenvolvimento mais confiável, rastreável e colaborativo.

6.2. Comparação do Processo IAP em Relação às Práticas Anteriores

A análise do processo IAP evidenciou que o esforço inicial investido foi recompensado por melhorias no alinhamento entre as equipes, na redução de retrabalho e na eficiência ao lidar com mudanças e organizar prioridades. Esta seção apresenta os principais resultados obtidos em quatro aspectos: a) o esforço necessário para compreender, codificar e testar; b) o esforço para estruturação; c) o impacto de pedidos de mudança; e d) a transferência de conhecimento e colaboração entre *stakeholders*. A combinação entre os dados qualitativos, obtidos nas entrevistas, e os dados quantitativos extraídos do questionário e das análises dos artefatos, fortaleceu a compreensão do impacto do processo.

6.2.1. Esforço Necessário para Compreender, Codificar e Testar

Antes da implantação do processo IAP, os testes de aceitação eram conduzidos com base unicamente na descrição textual das histórias de usuário, sem a definição formal de critérios de aceitação voltados ao cliente. Essa abordagem frequentemente gerava interpretações ambíguas por parte dos desenvolvedores e testadores, o que resultava em retrabalho, inconsistências nas entregas e

dificuldades na validação dos requisitos. A ausência de uma estrutura clara também dificultava a automação de testes e comprometia a rastreabilidade entre os requisitos e sua verificação.

Com a adoção do IAP, a equipe passou a sistematizar os testes a partir de critérios de aceitação bem definidos e estruturados, baseados em práticas de ATDD (*Acceptance Test-Driven Development*). Isso proporcionou maior clareza na comunicação entre os papéis da equipe, alinhamento com as expectativas dos *stakeholders* e, conseqüentemente, uma diminuição significativa de esforço na codificação e nos testes. A definição antecipada dos critérios de aceitação passou a orientar tanto a implementação funcional quanto a criação de cenários de teste mais precisos, o que reduziu erros e aumentou a confiabilidade das entregas.

Essa percepção foi evidenciada pelos dados qualitativos e quantitativos obtidos nas entrevistas. Quanto ao esforço para codificar, 66,7% dos entrevistados avaliaram como “inferior” ao das abordagens anteriores, e os demais como “igual”. Isso indica que, apesar do esforço inicial necessário para compreensão do processo, a clareza dos critérios de aceitação passou a facilitar a implementação do código. Um desenvolvedor relatou: *“Os critérios bem definidos exigem mais atenção durante o desenvolvimento, mas evitam muitos retrabalhos depois.”*

No que se refere ao esforço necessário para testar, 83,3% dos entrevistados consideraram que o processo IAP demandou esforço “inferior” em comparação a abordagens anteriores. O QA destacou que a sistematização dos critérios possibilitou testes mais diretos e eficazes, afirmando: *“A estrutura ajudou a reduzir erros e simplificar os testes.”* Outro membro da equipe comentou: *“O esforço foi maior no início, mas proporcionou resultados muito positivos ao longo do projeto.”*

Portanto, apesar do esforço inicial elevado para compreensão e adaptação ao novo processo, os dados demonstram que o IAP, ao integrar o ATDD com critérios de aceitação bem definidos, contribuiu para melhorar a qualidade dos requisitos, tornar a codificação mais precisa e simplificar o processo de testes, proporcionando ganhos de eficiência para toda a equipe.

6.2.2. Esforço Necessário para Estruturação

Antes da adoção do processo IAP, a estruturação do *backlog* ocorria de maneira menos sistematizada, com base em descrições informais das histórias de usuário e sem o suporte de uma ferramenta visual que organizasse o conhecimento de forma colaborativa. Esse cenário frequentemente resultava em desalinhamento entre os membros da equipe e *stakeholders*, baixa visibilidade das prioridades do produto e dificuldades na identificação de dependências e objetivos de negócio. O planejamento das *sprints* acabava, por vezes, comprometido pela falta de clareza e consenso sobre o escopo das entregas.

Com a introdução do PBB Canvas no contexto do IAP, a estruturação inicial do *backlog* passou a exigir um esforço maior de colaboração e tempo, especialmente durante as sessões de refinamento. Em relação ao custo de tempo para realização das especificações, observou-se, a partir da percepção

dos participantes, que a adoção do IAP demandou um aumento no tempo despendido nas etapas iniciais, principalmente na definição dos critérios de aceitação e na elaboração dos cenários em BDD. No entanto, esse esforço adicional foi percebido como compensatório ao longo do ciclo de desenvolvimento, uma vez que contribuiu para a redução de retrabalho, diminuição de dúvidas durante a implementação e maior assertividade na validação das funcionalidades.

Entretanto, essa abordagem trouxe benefícios: ao consolidar informações sobre o produto, como objetivos, problemas, expectativas, *personas*, funcionalidades e PBIs em uma estrutura visual e lógica, o Canvas passou a facilitar o entendimento compartilhado entre os papéis envolvidos. Isso resultou em maior alinhamento estratégico, melhor organização das demandas e mais fluidez no planejamento das sprints seguintes.

Essa percepção foi confirmada tanto pelas entrevistas quanto pelos dados quantitativos obtidos no questionário. Quando questionados sobre a estrutura da integração entre as práticas ATDD e PBB, 100% dos entrevistados atribuíram nota 5 (muito adequada), evidenciando um elevado nível de satisfação com a clareza e organização do processo. Esse resultado demonstra que, mesmo diante do esforço inicial necessário para adaptação, a equipe reconheceu o valor prático e estratégico da estrutura proposta.

Os relatos também refletem essa avaliação positiva. Um dos participantes afirmou: *“A estrutura promoveu clareza, colaboração e alinhamento, facilitando a gestão do backlog e o planejamento das sprints”*, evidenciando a percepção de valor agregado pela ferramenta. Outro destacou: *“A ferramenta proporcionou um alinhamento claro entre as partes interessadas, organizando as prioridades do produto de maneira lógica e colaborativa”*, reforçando o papel do PBB Canvas como facilitador da comunicação e da definição das entregas.

Assim, embora a fase inicial de estruturação com o PBB Canvas tenha exigido um investimento considerável de tempo e esforço, os ganhos obtidos na organização do *backlog*, no alinhamento das expectativas e na eficiência do planejamento compensaram amplamente o esforço despendido.

6.2.3. Análise de Impacto de Pedidos de Mudança

Antes da implantação do processo IAP, a gestão de mudanças era frequentemente conduzida de forma reativa, com solicitações surgindo em momentos avançados do ciclo de desenvolvimento. Isso dificultava a readequação das histórias de usuário e tarefas, gerando retrabalho e impactos no cronograma. A ausência de uma estrutura para documentar e analisar os critérios de aceitação tornava mais lenta a identificação de pontos críticos, aumentando o esforço para adaptar os requisitos e manter a consistência do *backlog*.

Com a adoção do IAP, a absorção de mudanças passou a ser feita com mais organização. O uso do PBB Canvas, aliado à definição estruturada dos critérios de aceitação com base no ATDD, favoreceu a rastreabilidade e a previsibilidade ao lidar com alterações. A visualização das prioridades e dos

vínculos entre os itens do *backlog* permitiu à equipe analisar melhor os impactos e tomar decisões mais informadas, embora ainda exigisse atenção constante para manter esse alinhamento ao longo das *sprints*.

As entrevistas indicaram que, mesmo com desafios, houve percepção de melhoria na gestão de mudanças. Um entrevistado apontou: “*O processo foi mais eficiente em absorver mudanças sem prejudicar o cronograma*”. Outro destacou: “*A organização do PBB Canvas facilitou a incorporação de mudanças*”, e um terceiro observou que “*A estrutura do ATDD reduziu os impactos das mudanças*”. Esses relatos demonstram que o modelo foi visto como uma forma mais controlada de lidar com as alterações, embora não tenha eliminado totalmente os efeitos dessas demandas no fluxo de trabalho.

Complementando a análise qualitativa, os dados quantitativos mostraram que 100% dos entrevistados classificaram o impacto das mudanças como “inferior” em comparação com abordagens anteriores. No entanto, é importante ressaltar que essa avaliação reflete a percepção da equipe no contexto específico da implantação e pode estar relacionada à fase de experimentação e maior atenção às práticas durante o período de aplicação.

Assim, embora os dados apontem vantagens na forma como as mudanças foram tratadas com o IAP, os resultados devem ser interpretados considerando as limitações do cenário observado. A gestão de mudanças ainda exige acompanhamento constante, e a estrutura do IAP demonstrou-se um apoio útil, mas não isento de ajustes para contextos diversos.

6.2.4. *Transferência de Conhecimento, Colaboração do Cliente e Dependência entre Stakeholders*

Antes da adoção do processo IAP, a colaboração entre os membros da equipe e os *stakeholders* era mais informal e pontual, com alinhamentos ocorrendo, na maioria das vezes, durante as reuniões de planejamento ou por meio de canais de comunicação não estruturados. Isso frequentemente gerava interpretações divergentes, desalinhamento nas expectativas e dificuldades na compreensão dos objetivos do produto. A transferência de conhecimento também ocorria de forma fragmentada, com baixa rastreabilidade das decisões tomadas ao longo das *sprints*.

Com a introdução do IAP, a colaboração entre os *stakeholders* tornou-se mais constante e eficaz. A utilização de ferramentas visuais, como o PBB Canvas, permitiu consolidar e comunicar informações relevantes de forma clara e acessível a todos os envolvidos. Além disso, os critérios de aceitação bem definidos facilitaram a troca de conhecimento técnico entre desenvolvedores, testadores e *Product Owners*, promovendo um entendimento compartilhado dos requisitos. No entanto, mesmo com esses avanços, foi identificado que uma participação mais ativa do cliente nas fases de refinamento poderia ter potencializado ainda mais os benefícios do processo.

Essa percepção ficou evidente nas falas dos entrevistados. Um deles afirmou: “*...garantiu uma comunicação clara entre as equipes, alinhando as expectativas do cliente e as entregas da equipe de desenvolvimento*”, indicando que a estrutura do IAP fortaleceu o alinhamento interno. Outro

complementou: “...foi superior ao que eu esperava, especialmente no alinhamento das expectativas entre o cliente e a equipe de desenvolvimento”, ressaltando que o processo superou as práticas anteriores nesse aspecto. Por outro lado, um dos participantes destacou uma oportunidade de melhoria ao dizer: “A participação do cliente foi regular, com reuniões semanais. Acho que poderia haver maior interação durante o refinamento do backlog”, sugerindo que o engajamento do cliente, embora presente, ainda poderia ser ampliado.

De modo geral, as entrevistas revelaram uma aceitação positiva em relação à aplicação dos critérios de aceitação e à estrutura do processo IAP. No entanto, foram identificadas áreas que demandam melhorias específicas para otimizar o fluxo. Entre os pontos destacados, os participantes sugeriram uma maior padronização na escrita dos critérios, utilizando uma linguagem mais orientada ao entendimento do cliente para facilitar a comunicação.

Outro ponto de atenção refere-se às ferramentas de apoio. Houve sugestões para melhorar a visualização dos artefatos gerados no PBB Canvas e garantir uma integração mais fluida com as ferramentas de gestão ágil (como o *Azure DevOps*) já adotadas pela equipe. Quanto à colaboração, reforçou-se a necessidade de uma atuação mais ativa do cliente durante os refinamentos, garantindo um alinhamento mais profundo entre a solicitação e a entrega final.

Assim, esta pesquisa acredita que o processo IAP contribuiu para a melhoria da comunicação e da transferência de conhecimento entre as partes interessadas, embora reste espaço para evoluir na promoção de uma colaboração mais próxima e contínua com o cliente ao longo de todo o ciclo de desenvolvimento.

6.3. Resultados da Avaliação dos Critérios de Pesquisa-Ação

Para avaliar a eficácia da pesquisa-ação conduzida, foram definidos três critérios principais, baseados em Thiollent (2007): a) promover entendimento comum entre os participantes, b) criar e assimilar conhecimento e c) avaliar o impacto no campo de desenvolvimento ágil de *software*. Esses critérios foram escolhidos por estarem alinhados aos objetivos centrais da pesquisa-ação: promover mudanças práticas no ambiente de trabalho, gerar conhecimento aplicável e mensurar os efeitos da intervenção no contexto estudado.

Quanto ao primeiro critério, que avalia se a pesquisa-ação promoveu um entendimento comum entre os participantes, em especial entre a equipe de desenvolvimento e os *stakeholders*, os dados apontam avanços consistentes. A avaliação partiu da pergunta: “Como é a colaboração entre os *stakeholders* da equipe de desenvolvimento?”, considerando também aspectos como frequência e adequação da interação. Segundo os dados quantitativos, 91,7% dos entrevistados classificaram a colaboração como “superior” em comparação a abordagens utilizadas anteriormente. Essa percepção foi atribuída principalmente ao uso de ferramentas como o PBB Canvas, que facilitou a visualização dos elementos do produto e promoveu um alinhamento mais efetivo entre as partes.

As entrevistas qualitativas reforçam esse dado, com participantes mencionando que “os critérios bem definidos garantiram maior alinhamento nas entregas” e “houve maior clareza na comunicação interna”. No entanto, também foi observado que esse entendimento comum não se estabeleceu de forma imediata. Alguns entrevistados relataram que o alinhamento entre os papéis da equipe e os *stakeholders* se fortaleceu progressivamente ao longo das *sprints*, indicando que o processo exigiu tempo de adaptação e amadurecimento coletivo.

Portanto, embora os dados mostrem uma colaboração mais eficaz com a adoção do IAP, os relatos também revelam que o entendimento comum foi resultado de um esforço gradual, e não de uma mudança instantânea. Isso reforça a importância de ciclos iterativos e momentos regulares de *feedback* para consolidar a colaboração no ambiente ágil.

O segundo critério buscou avaliar se a pesquisa-ação promoveu a criação e assimilação de novo conhecimento pelos participantes. A adoção prática de artefatos como o PBB Canvas e a formalização dos critérios de aceitação demonstraram-se eficazes nesse aspecto. Relatos como “A implementação do processo facilitou a adoção de práticas que antes eram difíceis de integrar ao dia a dia” indicam que os participantes internalizaram e aplicaram os aprendizados no contexto real. De forma complementar, 83,3% dos entrevistados classificaram a assimilação de conhecimento como “superior”, demonstrando que o processo contribuiu para fortalecer a aprendizagem organizacional. Ainda assim, alguns participantes apontaram que o ritmo de aprendizado foi mais lento entre membros com menor familiaridade prévia com metodologias ágeis, revelando a necessidade de reforço em capacitações iniciais.

O terceiro critério avaliou o impacto do processo no campo do desenvolvimento ágil, com foco na melhoria da especificação de requisitos. A percepção geral foi positiva, com 100% dos entrevistados considerando que o processo contribuiu para maior previsibilidade e redução de retrabalho. Um participante destacou: “O impacto das mudanças foi menor devido à organização proporcionada pelo processo”, enquanto outro observou: “O alinhamento com as melhores práticas trouxe mais confiança no planejamento”. No entanto, parte dos entrevistados indicou que a adoção total da abordagem exigiu ajustes no ritmo das cerimônias, especialmente durante o refinamento, o que sinaliza que os benefícios ocorreram de forma gradual.

De maneira geral, os resultados apontam que os critérios de pesquisa-ação, relevância, aplicabilidade prática e impacto positivo no contexto analisado, foram amplamente atendidos. Entretanto, é importante destacar que os ganhos observados ocorreram dentro de um cenário controlado, em um período limitado (três *sprints*) e com um time já parcialmente familiarizado com práticas ágeis. Isso sugere que, embora os resultados tenham sido positivos, a replicação em outros contextos exigirá adequações, especialmente em equipes com menos maturidade ou maior rotatividade.

6.4. Síntese Temática dos Resultados

A síntese temática dos resultados obtidos após a aplicação do processo IAP sugeriu aspectos centrais que impactaram positivamente o desempenho e a organização da equipe. Essa seção apresenta uma visão consolidada dos principais temas identificados, destacando a clareza nos requisitos, a colaboração aprimorada entre os membros da equipe, os benefícios no cumprimento do cronograma e as melhorias na qualidade das entregas.

Seguindo com o método proposto por Cruzes e Dyba (2011), conforme descrito na seção 3.2, as etapas seguidas do processo foram as seguintes:

- **Leitura Preliminar dos Dados:** A leitura inicial das entrevistas revelou padrões recorrentes nas percepções dos participantes sobre a integração das práticas ATDD e PBB. Foram mencionadas tanto as contribuições do processo, como a melhora na clareza dos requisitos e na organização do *backlog*, quanto os desafios enfrentados na adaptação inicial, principalmente relacionados à carga de trabalho e à necessidade de capacitação contínua. Também emergiram pontos sobre colaboração entre os membros da equipe e stakeholders, com destaque para a comunicação mais fluida. Apesar dos aspectos positivos, os dados também indicaram que os resultados ocorreram de forma gradual, e que o processo ainda demanda ajustes para maior efetividade.

- **Identificação e Rotulação de Segmentos (Códigos):**
 - **Eficácia do Processo:**
 - *"Clareza nos Critérios de Aceitação"*: O entrevistado relatou à redução de ambiguidades nos testes e no desenvolvimento.
 - *"Organização Através do PBB Canvas"*: Entrevistados valorizaram como a ferramenta estruturou e consolidou informações.

 - **Desafios de Adaptação:**
 - *"Esforço Inicial Elevado"*: Um entrevistado mencionou o impacto do detalhamento exigido e da curva de aprendizado.
 - *"Necessidade de Capacitação Contínua"*: Entrevistado sugeriu treinamentos regulares para adoção plena das práticas.

 - **Facilitação na Colaboração e Comunicação:**
 - *"Participação do Cliente"*: Os entrevistados elogiaram a frequência e adequação da interação com os *stakeholders*.

- *"Alinhamento Entre as Equipes"*: Entrevistado elogiou como a abordagem melhorou a comunicação entre desenvolvedores, QA e POs.
- **Oportunidade de Melhorias:**
 - *"Impacto Positivo nas Mudanças"*: Entrevistados valorizaram a eficiência na absorção de alterações de requisitos.
 - *"Possibilidade de Integração com Outras Ferramentas"*: O entrevistado sugeriu o uso de ferramentas adicionais para potencializar os resultados do processo.
- **Agrupamento de Códigos em Temas:**
 - **Processo**: Abrange códigos sobre a clareza trazida pelos critérios de utilização e organização do PBB Canvas.
 - **Adaptação**: Inclui códigos que destacam sobre os esforços para aprender e aplicar as práticas ATDD e PBB e treinamentos para adoção do processo.
 - **Colaboração e Comunicação**: Envolve códigos que indicam uma interação entre equipes e *stakeholders* e uma melhora na comunicação entre o time.
 - **Melhorias**: Reflete códigos relacionados à eficiência no gerenciamento de alterações durante o projeto e ferramentas adicionais que potencialize os resultados do processo.
- **Identificação de Categorias Abrangentes**: Os temas foram consolidados em quatro categorias abrangentes que refletem os resultados gerais da pesquisa:
 - **Eficácia do Processo**: Os critérios de aceitação e a estrutura do PBB Canvas facilitaram a especificação de requisitos, reduzindo ambiguidades. Segundo os dados, 91,7% dos entrevistados avaliaram a comunicação como superior em comparação às abordagens anteriores.
 - **Desafios de Adaptação**: O esforço inicial foi destacado como um ponto crítico. Embora todos tenham conseguido se adaptar ao longo das sprints, 75% relataram que o nível de detalhamento exigido inicialmente foi maior do que estavam habituados, reforçando a necessidade de capacitação contínua.
 - **Facilitação na Colaboração e Comunicação**: Os dados qualitativos revelam melhora significativa na interação entre os papéis da equipe, enquanto os dados quantitativos apontam que 83,3% dos entrevistados consideraram a colaboração com *stakeholders* como adequada ou superior às experiências anteriores.
 - **Oportunidade de Melhorias**: Embora o processo tenha se mostrado útil na absorção de mudanças e organização das tarefas, os participantes sugeriram a integração com ferramentas já utilizadas pelo time e a automatização de etapas como pontos de aprimoramento.

Essa análise integrada permitiu compreender a aplicação do processo IAP de forma ampla e crítica, reconhecendo seus impactos positivos, suas limitações e as possibilidades de evolução em contextos futuro.

6.5. Reflexões sobre os resultados

A integração entre ATDD e PBB apoiou o desenvolvimento de *software* ágil, tendo o potencial para contribuir com o alinhamento de requisitos, a colaboração entre equipes e a eficiência do processo de desenvolvimento, conforme discutido a seguir:

- **Alinhamento com os Requisitos de Negócio:** O ATDD tem como premissa a criação de testes de aceitação antes do início do desenvolvimento, com base em critérios estabelecidos pelos *stakeholders*. Essa prática contribui para que as funcionalidades implementadas estejam diretamente ligadas às expectativas e necessidades do negócio. Paralelamente, o PBB é uma técnica voltada para a construção de um *backlog* de produto estruturado e priorizado, de forma a refletir as funcionalidades mais importantes e o valor de negócio a ser entregue. Ao unir essas duas práticas, os itens do *backlog* são definidos com mais clareza e baseados em critérios de aceitação bem estabelecidos. Isso permite que as funcionalidades sejam validadas desde o início, mitigando o risco de retrabalho ou de desenvolvimento de funcionalidades mal compreendidas.
- **Comunicação e Colaboração entre Equipes:** Um dos pilares do ATDD é a colaboração entre desenvolvedores, testadores e *stakeholders* na definição dos critérios de aceitação. Essa colaboração promove um entendimento compartilhado dos requisitos e reduz a possibilidade de interpretações divergentes. Da mesma forma, o PBB potencializa que as prioridades do *backlog* sejam alinhadas com o *Product Owner* e a equipe de desenvolvimento, criando um entendimento comum sobre o que deve ser desenvolvido. A combinação dessas práticas resulta em um processo de desenvolvimento colaborativo, onde todos os envolvidos têm uma visão clara de como cada funcionalidade será testada e validada. Isso aprimora a comunicação entre as equipes e ajuda que o trabalho esteja alinhado com as expectativas de negócio.
- **Redução de Retrabalho:** Ao utilizar o ATDD, os critérios de aceitação são definidos antecipadamente, o que previne ambiguidades e mudanças tardias nos requisitos. O PBB permite que o *backlog* seja organizado de forma incremental e adaptável, priorizando as funcionalidades que agregam mais valor ao cliente. Integrar o ATDD no processo de construção do *backlog* possibilita identificar e alcançar uma solução para problemas de entendimento dos requisitos ainda nas fases iniciais. Dessa forma, minimiza-se o retrabalho, pois os critérios de aceitação e os requisitos do *backlog* são validados antes do início do desenvolvimento.

- **Ciclos de *Feedback* mais curtos e Iterações mais ágeis:** O ATDD possibilita que os testes de aceitação sejam executados ao longo do processo de desenvolvimento, oferecendo *feedback* contínuo sobre o progresso do trabalho. Por sua vez, o PBB organiza o *backlog* de forma a permitir iterações constantes e entregas incrementais de valor. Ao integrar as duas abordagens, os ciclos de *feedback* se tornam mais curtos, pois os testes de aceitação são aplicados diretamente nas funcionalidades priorizadas e iteradas no *backlog*. Isso possibilita ajustes rápidos, seja em função de mudanças nos requisitos, seja pela necessidade de realinhamento de prioridades, tornando o processo de desenvolvimento mais ágil.
- **Entrega de Valor de Negócio:** O ATDD intensifica que cada funcionalidade desenvolvida seja orientada a resultados tangíveis e validada por meio de critérios de aceitação. O PBB organiza o *backlog* de maneira a potencializar que as funcionalidades de maior valor sejam priorizadas para entrega. A combinação das duas práticas possibilita que o desenvolvimento seja conduzido com foco não apenas na implementação de funcionalidades, mas na entrega de valor ao cliente. As funcionalidades desenvolvidas são validadas com base em critérios de aceitação bem definidos, e as entregas estão alinhadas com as prioridades estabelecidas no *backlog*, minimizando o risco de desenvolver funcionalidades que não atendam aos objetivos do negócio.
- **Aumento da Visibilidade e Transparência no Processo:** No ATDD, os testes de aceitação são escritos e visíveis a todos os envolvidos no projeto, proporcionando maior transparência. O PBB cria uma visão clara das prioridades e do status de cada item do *backlog*, permitindo que todos os *stakeholders* acompanhem o progresso do projeto. Quando essas abordagens são integradas, a equipe de desenvolvimento e os *stakeholders* têm uma visão clara e objetiva do que está sendo desenvolvido, como será testado e quando será entregue. Essa transparência facilita o gerenciamento de expectativas e a tomada de decisões durante o desenvolvimento.

Apesar dos desafios iniciais na implantação, acredita-se que o processo tenha potencial para contribuir com as atividades de requisitos e testes em equipes ágeis. As lições aprendidas e as sugestões dos participantes podem servir de base para refinamentos futuros e para a aplicação em outros projetos.

7. CONCLUSÃO

Este capítulo apresenta uma síntese das principais construções realizadas ao longo do trabalho, revisitando os objetivos propostos, bem como os resultados alcançados com a investigação sobre a integração das práticas ATDD e PBB no contexto do desenvolvimento ágil de *software*.

7.1. Contribuições da Pesquisa

Ao longo da pesquisa, foi possível desenvolver e implantar um processo sistematizado, o IAP, em um ambiente real de projeto, o que permitiu observar contribuições práticas e teóricas, ao mesmo tempo em que se reconheceram limitações e desafios inerentes ao seu uso.

A pesquisa teve início com a identificação de lacunas na literatura e na prática de especificação de requisitos em ambientes ágeis. O Capítulo 2 apresentou os fundamentos teóricos que sustentam as práticas ATDD e PBB, destacando seus potenciais de complementaridade. O capítulo 3 abordou uma avaliação do processo de desenvolvimento de software de uma empresa, como estratégia para identificar requisitos para a integração do ATDD e PBB. No Capítulo 4 foi proposto o processo IAP para integrar essas práticas de forma sistemática e orientada à realidade da equipe envolvida.

No Capítulo 5 foi descrita a implantação do processo que foi realizada na mesma empresa durante três *sprints*, com um time composto por desenvolvedores, QAs, POs e gerente de projeto, utilizando a metodologia *Scrum* e a ferramenta *Azure DevOps*. A capacitação inicial, o monitoramento contínuo e as dificuldades enfrentadas foram discutidas com base nas experiências reais dos participantes. As evidências obtidas, incluindo a alta taxa de completude e legibilidade dos requisitos avaliados, apontaram que, embora o esforço de adaptação tenha sido significativo, houve melhora na organização do *backlog*, clareza nos critérios de aceitação e maior alinhamento entre os papéis da equipe.

O Capítulo 6 apresentou a avaliação do processo implantado, com base em roteiro de entrevistas. Os dados analisados mostraram que o IAP contribuiu para a clareza dos requisitos, melhor comunicação interna e resposta mais eficiente às mudanças. A comparação com práticas anteriores indicou percepções positivas quanto à estrutura, aos critérios de aceitação e à colaboração com *stakeholders*, embora também tenham sido sugeridas melhorias, como maior capacitação prévia e integração com outras ferramentas.

Esta pesquisa traz contribuições para o campo do desenvolvimento ágil de *software*, especialmente no contexto da engenharia de requisitos. A proposta de integração das práticas ATDD e PBB resultou em um processo sistemático e estruturado chamado IAP, que buscou diminuir ou resolver algumas lacunas identificadas na literatura e na prática de desenvolvimento de software. As principais contribuições estão relacionadas aos seguintes aspectos:

- **Especificação de Requisitos atrelada aos testes:** A integração das práticas permitiu uma abordagem mais clara e colaborativa na especificação de requisitos. Os critérios de aceitação bem definidos contribuíram para a redução de ambiguidades e para a criação de um *backlog* mais alinhado às expectativas do cliente.
- **Fortalecimento da Colaboração entre Equipes:** A pesquisa demonstrou que a colaboração entre desenvolvedores, testadores e *stakeholders* tende a ser aprimorada com a adoção do Processo IAP. As ferramentas utilizadas, como o PBB Canvas, facilitaram o alinhamento e a comunicação entre as partes interessadas, promovendo um entendimento compartilhado dos objetivos do projeto.
- **Contribuição Prática:** A implantação do processo IAP em um ambiente real de desenvolvimento foi importante para validar as ações propostas e identificar oportunidades de melhoria. O processo tem o potencial de ser replicável em outros contextos, desde que sejam realizadas as devidas adaptações às particularidades de cada equipe ou organização.

Por fim, esta pesquisa reforça a importância de integrar práticas ágeis de forma estruturada para enfrentar desafios relacionados à engenharia de requisitos, contribuindo não apenas para o campo acadêmico, mas também para a indústria de *software*. As experiências, desafios e resultados obtidos durante a implantação da abordagem, fornecem subsídios importantes para futuras pesquisas e organizações que busquem a implementação do processo IAP.

7.2. Trabalhos Futuros

Apesar dos avanços observados na aplicação do processo IAP, a pesquisa apresentou limitações que abrem espaço para investigações complementares. Os resultados obtidos fornecem subsídios iniciais, mas ainda demandam aprofundamentos para que o processo seja compreendido e validado em maior amplitude. Algumas possibilidades para trabalhos futuros incluem:

- **Avaliação de Longo Prazo:** Estudos com aplicação do processo em períodos mais extensos poderiam permitir uma observação mais consistente dos efeitos, dificuldades e adaptações ao longo do tempo.
- **Adaptação a Diferentes Contextos:** É importante explorar o desempenho do processo em equipes de diferentes portes, com distintos níveis de maturidade ágil ou atuando em domínios variados, a fim de avaliar sua flexibilidade e necessidade de ajustes.
- **Automatização dos Critérios de Aceitação:** Novos estudos podem explorar formas de automatizar a elaboração e execução dos critérios de aceitação, considerando o potencial de ganho de produtividade e consistência no processo.

- **Participação do Cliente:** Identificou-se que a presença do cliente foi relevante, mas ainda pontual. Pesquisas futuras podem aprofundar estratégias para tornar essa participação mais efetiva, especialmente nas etapas de refinamento e validação.

Essas direções não visam validar ou invalidar o processo proposto, mas sim contribuir para seu amadurecimento e adaptação a diferentes realidades de desenvolvimento ágil, permitindo que suas aplicações sejam mais bem compreendidas e ajustadas à prática.

7.3. Limitações da pesquisa e Ameaças à Validade

A validação dos resultados obtidos com a aplicação do processo de integração ATDD e PBB requereu uma análise criteriosa das possíveis ameaças que podiam comprometer a interpretação e generalização das conclusões. Essa avaliação abrangeu aspectos relacionados ao contexto da pesquisa, às limitações na coleta de dados e ao próprio envolvimento dos participantes.

Primeiramente, a possibilidade de viés nas respostas devido ao envolvimento direto dos participantes no projeto foi uma das ameaças identificadas. Como a equipe estava ativamente envolvida na aplicação das práticas avaliadas, haveria o risco de que as percepções fossem influenciadas pelo desejo de demonstrar resultados positivos. Embora isso tenha sido mitigado pelo uso de entrevistas semiestruturadas e escalas padronizadas, a subjetividade permaneceu como um fator a ser considerado.

Outra limitação estava no tempo disponível para a avaliação. O estudo foi realizado ao longo de três *sprints*, o que poderia não ser suficiente para capturar o impacto de longo prazo das mudanças implementadas. Processos como a integração de novas práticas, frequentemente apresentam benefícios que se consolidam apenas após períodos mais extensos de aplicação, o que sugeriu a necessidade de estudos futuros para validar os achados em contextos de maior duração.

O número de participantes entrevistados também foi um fator que pode influenciar a validade externa dos resultados. Embora as percepções coletadas tenham sido ricas e diversificadas, a amostra limitada a uma única equipe reduz a capacidade de generalizar as conclusões para outros cenários ou organizações. Estudos adicionais em diferentes contextos poderiam oferecer maior robustez às descobertas.

Além disso, o contexto específico do projeto e as ferramentas utilizadas podem limitar a replicabilidade dos resultados. O uso do PBB Canvas e dos critérios de aceitação, por exemplo, mostrou-se altamente eficaz nesse caso, mas sua aplicabilidade pode variar dependendo das características da equipe ou do ambiente organizacional. Avaliações adicionais em equipes com diferentes níveis de maturidade ágil seriam úteis para explorar essas variações.

Por fim, foi observado que a participação do cliente, embora presente, poderia ter sido mais ativa em alguns momentos críticos do processo, como o refinamento do *backlog*. Essa interação limitada pode ter impactado a abrangência do alinhamento entre as necessidades do cliente e as entregas

da equipe. Aumentar a frequência e a profundidade dessa colaboração é uma recomendação para trabalhos futuros, visando fortalecer ainda mais os resultados obtidos.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGUIAR, F. & CAROLI, P. (2021). *Product Backlog Building: Um guia prático para criação e refinamento de backlog para produtos de sucesso*. Editora Caroli.
- AGILE ALLIANCE. *Subway Map to Agile Practices*. 2025. Disponível em: <https://agilealliance.org/agile101/subway-map-to-agile-practices/>. Acesso em: 9 nov. 2025.
- BECK, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- BOURQUE, Pierre; FAIRLEY, Richard E. (Ed.). *SWEBOK: guide to the software engineering body of knowledge*. IEEE Computer Society, 2014.
- COAD, P., DE LUCA, J. (1997). *Feature Driven Development*. Disponível em: <http://www.featuredrivendevelopment.com/>. Acesso em: mar. 2024.
- COHN, Mike. *Desenvolvimento de software com Scrum: Aplicando métodos ágeis com sucesso*. Porto Alegre, RS: Bookman, 2011.
- COUGHLAN, P.; COUGHLAN, D. Action research for operations management. *International Journal of Operations & Production Management*, v. 22, n. 2, p. 220-240, 2002.
- CORRÊA, G. C. G., DE CAMPOS, I. C. P., & ALMAGRO, R. C. (2018). Pesquisa-ação: uma abordagem prática de pesquisa qualitativa. *Ensaios pedagógicos*, 2(1), 62-72.
- CRUZ, Fábio. *Scrum e Agile em projetos*. 2018, 1ed. Rio de Janeiro, RJ: BRASPORT, 2018.
- CRUZES, Daniela S.; DYBA, Tore. Recommended steps for thematic synthesis in software engineering. In: *2011 international symposium on empirical software engineering and measurement*. IEEE, 2011. p. 275-284.
- DA SILVA¹, Bleno WFV; OLIVEIRA¹, Sandro RB. *REACT-M: O Relato de um Estudo de Caso de Aplicação de um Método Ágil para Gerência de Requisitos de Software*. 2020.
- DARAOJIMBA, Emmanuel Chibuike et al. Comprehensive review of agile methodologies in project management. *Computer Science & IT Research Journal*, v. 5, n. 1, p. 190-218, 2024
- FAGUNDES, Priscila Basto et al. Taxonomias, ontologias e tesouros: possibilidades de contribuição para o processo de Engenharia de Requisitos. *Em Questão*, p. 237-254, 2020.
- FERREIRA, Bruna et al. Investigating Problem Definition and End-User Involvement in Agile Projects that Use Lean Inceptions. In: *Proceedings of the XX Brazilian Symposium on Software Quality*. 2021. p. 1-10.
- FIGUEIRA, Anderson Marques da Silva. *Análise das técnicas de levantamento de requisitos para desenvolvimento de software nas empresas de Vitória da Conquista – BA*, 2012.
- HOFFMANN, Luiz Felipe Simoes et al. Applying acceptance test driven development to a problem-based learning academic real-time system. In: **2014 11th International Conference on Information Technology: New Generations**. IEEE, 2014. p. 3-8
- INAYAT, Irum et al. A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, v. 51, p. 915-929, 2015.
- LISBÔA, D. A., DA ROCHA, T. A., MACHADO, L. S., CALDEIRA, C. M., & DE SOUZA, C. R. (2021, April). Um Estudo Observacional sobre as Adaptações ao Trabalho Remoto no contexto

da Pandemia de COVID-19. In Anais do XVII Simpósio Brasileiro de Sistemas Colaborativos (pp. 95-106). SBC.

LONGO, Douglas Hiura et al. Métricas para medir a uniformidade de dados em testes de aceitação. 2020.

MASSARI, Vitor L. Gerenciamento ágil de projetos. 2018, 2ed. Rio de Janeiro, RJ: BRASPORT, 2018. ISBN: 9788574528939.

MEDEIROS, J. D. R. V. de; ALVES, D. C. P.; VASCONCELOS, A. M. L. de; SCHUENEMANN, C. T. L. L. S.; WANDERLEY, E. Engenharia de requisitos em projetos ágeis: uma revisão sistemática da literatura. Revista Principia, [S. l.], v. 1, n. 28, p. 11–24, 2015. DOI: 10.18265/1517-03062015v1n28p11-24.

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, 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.

MCKAY, J.; MARSHALL, P. The dual imperatives of action research. Information technology and people. v. 14, p. 46–59, 2001.

MIGUEL, P.A.C., FLEURY, A., MELLO, C.H.P., NAKANO, D.N., LIMA, E.P. de L., TURRIONI, J.B., HO, L.L., MORABITO, R., MARTINS, R.A., SOUSA, R., COSTA, S.E.G., & PUREZA, V. (2012). Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações (2nd ed.). Elsevier Editora Ltda.

MIRANDA, Andrey Amaral; OLIVEIRA, Sandro Ronaldo Bezerra. Um relato de aplicação dos métodos ágeis REACT e REACT-M para o desenvolvimento e o gerenciamento dos requisitos de um software de streaming de vídeo. CONTRIBUCIONES A LAS CIENCIAS SOCIALES, v. 17, n. 4, p. e6325-e6325, 2024.

MOE, Myint Myint. Comparative study of test-driven development tdd, behavior-driven development bdd and acceptance test–driven development atdd. International Journal of Trend in Scientific Research and Development, v. 3, n. 4, p. 231-234, 2019.

MUNIZ, A., BOAS, C.V., CABRAL, B., COLARES, R. (2019). Jornada ágil da qualidade: aplique práticas no início do ciclo para potencializar a implantação contínua de software com qualidade. Editora Brasport.

MUNIZ, Antonio et al. Jornada DevOps: Unindo Cultura ágil, Lean e tecnologia para entrega de software com qualidade. Brasport, 2019.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de software-9. McGraw Hill Brasil, 2021.

REBELO, Paulo. Acceptance Test-Driven Development (ATDD), passo a passo. InfoQ. Disponível em: <https://www.infoq.com/br/articles/atdd-passo-a-passo/> . Acessado em 09 jun. 2025

ROCHA, F. G.; MISRA, S.; SOARES, M. S. Guidelines for Future Agile Methodologies and Architecture Reconciliation for Software-Intensive Systems. Electronics 2023, 12, 1582 [em linha]. 2023.

SALES, Matheus Nascimento et al. Estratégia para nortear o processo de engenharia de requisitos aplicada à metodologia ágil Scrum. Revista de Ciência da Computação, v. 4, n. 1, p. 13-27, 2022.

SOMMERVILLE, I. Engenharia de software. 10a Edição. Pearson Education, 2018.

SWAMINATHAN, Vasu. ATDD – An applied approach to continuous delivery.

Disponível em: <https://www.aspiresys.com/articles/atdd-applied-approach-to-continuous-delivery.pdf> Acessado em: 14 set. 2024.

THIOLLENT, M. Metodologia da pesquisa-ação. 14a Edição. São Paulo: Cortez Editora, 2005.

THIOLLENT, M. Metodologia da pesquisa-ação. 15. ed. São Paulo: Cortez, 2007.2012 (pp. 1-4). IEEE.

WIEGERS, Karl; BEATTY, Joy. Software requirements. Pearson Education, 2013.

APÊNDICES

APÊNDICE A – ROTEIRO DA ENTREVISTA PARA COLETA DE DADOS ANTES DA INTRODUÇÃO AO PROCESSO IAP

Apresentação

- Saudações e apresentação.
- Agradeça ao participante.

Introdução

O objetivo deste questionário é realizar uma pesquisa para avaliar como as atividades de requisitos estão sendo realizadas em um ambiente ágil da empresa.

Ficáramos 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	
Nome	
ID	Questões
Q1	Qual a sua função?
Q2	Qual é a sua formação acadêmica?
Q3	Quantos anos de experiência você tem nesta função?
Q4	Quantos anos de experiência com requisitos de software?
Q5	Quantos anos de experiência com métodos ágeis?

Agora falaremos sobre o conceito de qualidade da Especificação de Requisitos de Software (ERS) em projetos ágeis.

Conceito de qualidade de ERS	
ID	Questões
Q6	O que você considera uma Especificação de Requisitos de Software (ERS) com qualidade? Ou seja, para codificar um recurso, quais atributos (características) você gostaria de encontrar na ERS? Sonda: Por quê?
Q7	Quais fatores afetam a qualidade da ERS? Sonda: Por quê?

Q8	<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 ou em desenvolvimento ágil • Estrutura de documentação utilizada pelo projeto • A falta de ferramenta de apoio para automatizar as atividades de requisitos • Um ERS mais focado no cliente do que na equipe de desenvolvimento • Falta do entendimento compartilhado da necessidade do produto • Falta de priorização para itens de maior importância para o produto e maior valor para o cliente • Estrutura da granularidade dos itens do <i>backlog</i> • Mudanças nas regras de negócio • A falta da Equipe de Qualidade no momento do levantamento de requisitos <p>Sonda: Por quê?</p>
Q9	<p>Você acha que uma ERS dividida em vários artefatos facilita o entendimento ou é mais produtivo para o desenvolvedor obter informações de forma consolidada a partir de poucos artefatos?</p> <p>Sonda: Por quê?</p>

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

Conteúdo da ERS	
ID	Questões

Q10	<p>No projeto atual, quais artefatos são usados pelos desenvolvedores para codificar uma funcionalidade?</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>
Q11	<p>Como esses artefatos ajudam o desenvolvedor a codificar um requisito?</p> <p>Sonda: Por quê?</p>
Q12	<p>Como esses artefatos ajudam o analista de qualidade/ testador a testar um requisito?</p> <p>Sonda: Por quê?</p>
Q13	<p>Que outras informações não estão presentes na ERS usada atualmente, mas são necessárias para codificar um requisito?</p> <p>Sonda: Por quê?</p>
Q14	<p>Você conhece ou já trabalhou com a prática ágil <i>Acceptance Test-Driven Development</i> (ATDD) ou Desenvolvimento orientado a testes de aceitação?</p>
Q15	<p>Você conhece ou já trabalhou com a prática ágil <i>Product Backlog Building</i> (PBB) ou Construção do Backlog do Produto?</p>
Q16	<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 às perguntas?</p> <p>Sonda: Se a ERS não for suficiente, o que você acha que normalmente falta ao desenvolvedor?</p>
Q17	<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>

Q18	O que você considera que contribui para tornar a ERS desatualizada? Sonda: Por quê?
-----	--

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

Avaliação da Qualidade da Especificação Atualmente utilizada	
ID	Questões
Q19	No projeto atual, você considera que a ERS é ambígua, ou seja, pouco clara e sujeita a mais de uma interpretação? Sonda: Se SIM, o que você acha que contribui para que a ERS seja escrita dessa forma?
Q20	No projeto atual, você considera que a ERS apresenta problemas de consistência entre os requisitos ou entre os artefatos? Sonda: Se SIM, por que você acha que isso acontece?

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

Contexto do Processo de Desenvolvimento	
ID	Questões
Q21	Qual é a frequência de comunicação com o cliente?
Q22	Considera que a comunicação com o cliente é adequada ou compromete a execução das suas atividades? Sonda: Por quê?

Q23	Quais são as ferramentas usadas para apoiar as atividades de requisitos? Quais são os benefícios que elas fornecem?
Q24	Os testes são feitos diretamente da ERS, ou é necessário elaborar um documento específico apenas para os testes? Considera que os testes estão a serem feitos de forma adequada? Sonda: Por quê?

Q25	O projeto contém informações sobre as relações de dependência entre os requisitos? Sonda: Se sim, como essas informações são atualizadas ao longo do desenvolvimento?
-----	--

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	<p>Em geral, quanto tempo dura cada sprint (ciclo de desenvolvimento)? As validações de software com cliente são feitas ao final de cada sprint?</p> <p>Sonda: Se não, com que frequência?</p>
Q6	<p>Como é feita a análise de impacto quando são solicitadas mudanças nos requisitos de software?</p>
Q7	<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>
Q8	<p>Na sua opinião, quais são os fatores que afetam a qualidade da ERS?</p> <p>Sonda: Por quê?</p>
Q9	<p>Quais boas práticas poderiam ser utilizadas em projetos para melhorar o desempenho da equipe?</p> <p>Sonda: Como?</p>
Q10	<p>Os pagamentos pela prestação de serviços ao cliente estão relacionados com a funcionalidade entregue ou estes pagamentos são baseados em valores fixos independentemente do que está disponível em cada entrega?</p>

Obrigado!

Muito obrigado. Sua participação foi muito importante para esta pesquisa.

APÊNDICE B – TEMPLATE PARA CRIAÇÃO DE HISTÓRIA DE USUÁRIO

Título da História de Usuário:

(Descreva de forma resumida o objetivo da história)

Descrição da História de Usuário:

Como [tipo de usuário], eu quero [ação], para [resultado/benefício esperado].

Exemplo: *Como um usuário logado, eu quero visualizar meu histórico de pedidos, para acompanhar o status das minhas compras anteriores.*

Critérios de Aceite:

(Liste as condições que devem ser atendidas para que a história seja considerada concluída com sucesso)

1. [Critério de aceite 1]
2. [Critério de aceite 2]
3. [Critério de aceite 3]
4. [Critério de aceite 4]

Exemplo:

1. O usuário logado deve poder acessar seu histórico de pedidos na seção "Meus Pedidos".
 2. O histórico deve exibir pedidos ordenados por data de compra, do mais recente ao mais antigo.
 3. O usuário deve ver detalhes como número do pedido, data, status, valor total e itens comprados.
 4. Deve haver uma opção para filtrar os pedidos por status: "Todos", "Entregue", "Em processamento", "Cancelado".
-

Interface (Se Necessário):

(Descreva como a interface deve ser ou anexe um *mockup*/protótipo, se relevante)

Exemplo:

- A tela de "Histórico de Pedidos" deve conter um grid com as colunas: "Número do Pedido", "Data", "Status", "Valor Total" e "Ação".
 - Deve haver um *dropdown* para filtrar os pedidos por status.
 - No canto superior direito, deve haver um botão "Voltar" que leva o usuário à tela principal de "Minha Conta".
-

Dependências:

(Identifique se há dependências de outras histórias ou funcionalidades que precisam ser concluídas antes)

Exemplo:

- Esta história depende da conclusão da história "Cadastro de Pedido", que é responsável por inserir os dados de pedidos no sistema.

Refinamento com o Time:

(Use este espaço para anotações após o refinamento, ajustes ou novas informações adicionadas durante a reunião com o time de desenvolvimento e QA)

APÊNDICE C – TEMPLATE PARA CRIAÇÃO DO BDD – CENÁRIOS DE TESTE (QA)

(O QA insere aqui os cenários de teste cobrindo cada critério de aceite)

Cenário 1:

Nome do cenário: [Descrever brevemente o cenário]

Dado que [pré-condição ou estado inicial]

E [condição adicional, se aplicável]

Quando [ação ou evento realizado pelo usuário]

Então [resultado esperado ou resposta do sistema]

Cenário 2:

Nome do cenário: [Descrever brevemente o cenário]

Dado que [pré-condição ou estado inicial]

E [condição adicional, se aplicável]

Quando [ação ou evento realizado pelo usuário]

Então [resultado esperado ou resposta do sistema]

Cenário 3:

Nome do cenário: [Descrever brevemente o cenário]

Dado que [pré-condição ou estado inicial]

E [condição adicional, se aplicável]

Quando [ação ou evento realizado pelo usuário]

Então [resultado esperado ou resposta do sistema]

Anotações Finais:

(Espaço para anotações adicionais, se necessário)

Exemplo de Preenchimento:

História de Usuário:

Como um cliente, eu quero visualizar o histórico de pedidos, para acompanhar o status das minhas compras.

Critérios de Aceite:

1. O usuário deve poder visualizar o histórico de pedidos após realizar login.
 2. Os pedidos devem ser exibidos em uma lista ordenada por data.
 3. O sistema deve permitir filtrar pedidos por status (Entregue, Processando, Cancelado).
-

BDD - Cenários de Teste (QA):

Cenário 1: Exibir o histórico de pedidos corretamente

Dado que o usuário está logado no sistema

E acessa a página "Meus Pedidos"

Quando o usuário visualizar o histórico de pedidos

Então o sistema deve exibir todos os pedidos ordenados por data de compra, do mais recente ao mais antigo.

Cenário 2: Filtrar pedidos por status "Entregue"

Dado que o usuário está na página "Meus Pedidos"

E há pedidos com o status "Entregue"

Quando o usuário selecionar o filtro "Entregue"

Então o sistema deve exibir apenas os pedidos com o status "Entregue"

Cenário 3: Histórico de pedidos sem pedidos realizados

Dado que o usuário está logado no sistema

E acessa a página "Meus Pedidos"

Quando o usuário não tiver nenhum pedido realizado

Então o sistema deve exibir a mensagem "Nenhum pedido encontrado"

APÊNDICE D – ROTEIRO DA ENTREVISTA PARA AVALIAÇÃO DO PROCESSO IAP

Informação do respondente

Nome	
Email	
Função	

Introdução

O objetivo desta entrevista é avaliar a utilização do processo para integração das práticas ATDD e PBB no projeto, quais as melhores práticas, dificuldades e limitações do processo proposto. Acreditamos que este estudo é essencial para melhorar a adoção da abordagem na prática.

Ficáramos 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 as 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 para integração das práticas ATDD e PBB.

Aprendizagem da Abordagem	
ID	Questões
Q4	Como foi o processo de aprendizagem da integração das práticas ATDD e PBB?
Q5	O que te ajudou nesse processo de aprendizagem do processo? Sonda: Por quê?
Q6	O que te dificultou nesse processo de aprendizagem do processo? Sonda: Por quê?

As próximas questões são sobre a estrutura e o conteúdo da integração das práticas ATDD e PBB.

Estrutura e conteúdo da integração das práticas ATDD e PBB	
ID	Questões

Q7	<p>Como você avalia o PBB estruturado pela ferramenta Canvas, com conceitos de Product Name, Problems, Expectations, Personas, Features e PBI ?</p> <p>Sonda: Por quê?</p>
Q8	<p>Como você avalia o ATDD estruturado por critérios de aceitação, cenários de testes e testes automatizados?</p>
Q9	<p>Numa escala de um (inadequado) a cinco (muito adequado), como avalia a estrutura da integração das práticas ATDD e PBB?</p> <p>Sonda: Por quê?</p>
Q10	<p>Usando uma escala de um (inadequado) a cinco (muito adequado), como você avalia a conformidade do processo para integração das práticas ATDD e PBB 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>
Q11	<p>Quais as diferenças do processo para integração das práticas ATDD e PBB em relação ao que você utilizava anteriormente ?</p> <p>Sonda: O quê?</p>

Agora falaremos sobre como foram realizadas as atividades da equipe de

desenvolvimento utilizando o processo para integração das práticas ATDD e PBB.

Efeito do processo para integração das práticas ATDD e PBB no trabalho em equipe	
ID	Questões
Q12	Como você avalia a estratégia adotada para realizar os testes de aceitação? Sonda: Por quê?
Q13	Como você avalia o reaproveitamento de requisitos no projeto? Sonda: Por quê?
Q14	Como você avalia as ferramentas usadas para apoiar as atividades de requisitos? Sonda: Por quê?
Q15	Como é a participação do cliente no projeto? Sonda: Frequência? Adequado?
Q16	Como é a colaboração entre os <i>stakeholders</i> da equipe de desenvolvimento? Sonda: Frequência? Adequado?

Agora vamos falar sobre o esforço necessário para usar o processo para integração das práticas ATDD e PBB.

Esforço necessário para usar o processo para integração das práticas ATDD e PBB	
ID	Questões
Q17	Em relação ao esforço necessário, quais são as diferenças entre o processo para integração das práticas ATDD e PBB e outras abordagens que você usou antes?
Q18	Quais práticas utilizadas no projeto atrapalham seu desempenho? Sonda: Como?
Q19	Quais práticas utilizadas no projeto melhoram seu desempenho? Sonda: Como?

Q20	Usando uma escala discreta (muito, razoável, pouco), como você avalia o esforço necessário para usar processo para integração das práticas ATDD e PBB? Sonda: Por quê?
Q21	Usando uma escala discreta (inferior, igual, superior), como você avalia o esforço necessário para especificar usando o processo para integração das práticas ATDD e PBB em comparação com outras abordagens que você usou antes? Sonda: Por quê?
Q22	Usando uma escala discreta (inferior, igual, superior), como você avalia o esforço necessário para codificar a partir do processo para integração das práticas ATDD e PBB em comparação com outras abordagens que você usou antes? Sonda: Por quê?
Q23	Usando uma escala discreta (inferior, igual, superior), como você avalia o esforço necessário para testar o processo para integração das práticas ATDD e PBB em comparação com outras abordagens que você usou antes? Sonda: Por quê?
Q24	O esforço necessário para usar o processo para integração das práticas ATDD e PBB é diferente do que você esperava? Sonda: Por quê?

Agora vamos falar sobre as mudanças nos requisitos.

Mudanças nos requisitos	
ID	Questões
Q25	Usando uma escala discreta (ruim, normal, boa), como você avalia a estratégia adotada para analisar as solicitações de mudança nos requisitos? Sonda: Por quê?
Q26	Como as mudanças afetam seu trabalho?

Q27	Usando uma escala discreta (inferior, igual, superior), como você avalia o impacto dessas mudanças em comparação com outras abordagens que você usou anteriormente? Sonda: Por quê?
-----	--

Por fim, falaremos sobre oportunidades para melhorar a adoção do processo para integração das práticas ATDD e PBB.

Oportunidades de melhoria	
ID	Questões
Q28	Que mudanças você gostaria de fazer no processo para integração das práticas ATDD e PBB? Sonda: Por quê?
Q29	Quais são as práticas que você não recomenda usar em projetos futuros? Sonda: Por quê?
Q30	Quais práticas você recomenda usar em projetos futuros? Sonda: Por quê?

Muito obrigada. Sua participação foi muito importante para esta pesquisa.

ANEXOS

ANEXO A – TERMO DE CONSENTIMENTO DA PESQUISA

Título da Pesquisa: Processo Para Especificação de Requisitos através da Integração de ATDD e PBB

Pesquisadora: Sanmara Alves Araújo

Participante da Pesquisa e Representante da Empresa: Diego Marques

Prezado(a) participante:

Sou aluna do Programa de Mestrado em Tecnologia da Informação do Instituto Federal da Paraíba e atualmente estou conduzindo uma pesquisa sob a supervisão da Professora Doutora Juliana Dantas Ribeiro Viana de Medeiros e coorientação da Professora Doutora Nadja da Nóbrega Rodrigues. O objetivo da pesquisa é definir um processo para integrar de maneira sistemática as práticas ATDD (*Acceptance Test-Driven Development*) e PBB (*Product Backlog Building*) para especificar requisitos em projetos ágeis de software de forma a facilitar o entendimento da especificação de requisitos pelos desenvolvedores e a atuação das atividades de testes.

A sua contribuição será crucial para a coleta de dados que guiarão este estudo, relacionando aspectos da sua experiência profissional com a situação atual da empresa, procurando investigar a forma como as atividades de requisitos estão sendo realizadas, buscando obter informações necessárias que sirvam de insumos para a definição do processo. Para atingir esse objetivo, será implantado o processo proposto na empresa (ATDD e PBB), visando melhorar as especificações de requisitos, em seqüência, utilizamos vários métodos de coleta de dados, incluindo entrevistas estruturadas e não estruturadas, observação participante e não participante e discussões em reuniões para identificar o impacto nas atividades de especificação de requisitos, detalhando os pontos fortes e as limitações, de forma a contribuir e refinar o processo apresentado.

A participação neste estudo é completamente opcional e voluntária, portanto, se em algum momento você 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 da empresa serão mantidas em sigilo absoluto, e todas as informações que possam identificá-lo serão omitidas.

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

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

Quaisquer dúvidas relativas à pesquisa poderão ser esclarecidas:

Pesquisador	
E-mail	sanmara.alves@academico.ifpb.edu.br
Telefone	+55 83 99960-7295

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

Coorientadora	
E-mail	nadja.rodrigues@ifpb.edu.br
Telefone	+55 83 99309-9294

Após a apresentação dessas informações, pedimos sua permissão livre e voluntária para participar desta pesquisa.

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

Assinatura do Participante da Pesquisa

Assinatura da Pesquisadora

Assinatura da Orientadora

Assinatura da Coorientadora