



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS RUSSAS**  
**CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE**

**ANTONIO JULIANO FERNANDES ROCHA PINHEIRO**

**PROPOSTA DE OTIMIZAÇÃO DO MODELO DE PROCESSO DE UMA EMPRESA  
JÚNIOR PELA UTILIZAÇÃO DA ABORDAGEM HÍBRIDA SCRUMBAN**

**RUSSAS**

**2025**

ANTONIO JULIANO FERNANDES ROCHA PINHEIRO

PROPOSTA DE OTIMIZAÇÃO DO MODELO DE PROCESSO DE UMA EMPRESA  
JÚNIOR PELA UTILIZAÇÃO DA ABORDAGEM HÍBRIDA SCRUMBAN

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de software do Campus Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de software.

Orientadora: Prof. Ms. Valéria Maria da Silva Pinheiro.

RUSSAS

2025

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

P718p Pinheiro, Antonio Juliano Fernandes Rocha.  
Proposta de otimização do modelo de processo de uma empresa júnior pela utilização da abordagem híbrida Scrumban / Antonio Juliano Fernandes Rocha Pinheiro. – 2025.  
69 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2025.  
Orientação: Prof. Me. Valéria Maria da Silva Pinheiro.

1. Metodologia híbrida. 2. Scrumban. 3. Empresa Júnior. 4. Práticas ágeis. I. Título.

CDD 005.1

---

ANTONIO JULIANO FERNANDES ROCHA PINHEIRO

PROPOSTA DE OTIMIZAÇÃO DO MODELO DE PROCESSO DE UMA EMPRESA  
JÚNIOR PELA UTILIZAÇÃO DA ABORDAGEM HÍBRIDA SCRUMBAN

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de software do Campus Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de software.

Aprovada em: 07/08/2025.

BANCA EXAMINADORA

---

Prof. Ms. Valéria Maria da Silva  
Pinheiro (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof. Dra. Anna Beatriz dos Santos Marques  
Universidade Federal do Ceará (UFC)

---

Prof. Dra. Patricia Freitas Campos de Vasconcelos  
Universidade Federal do Ceará (UFC)

## **AGRADECIMENTOS**

Gostaria de agradecer, primeiramente, a Deus, por me conceder força, discernimento e serenidade para alcançar mais uma conquista e encerrar um ciclo tão importante na minha vida.

Agradeço, também, àqueles que sempre estiveram ao meu lado, acompanhando meu crescimento desde a infância — minha família. Em especial, meus pais e meus irmãos, que foram e continuam sendo meu alicerce. Sem o amor, o apoio e os valores que me transmitiram, certamente eu não teria me tornado a pessoa que sou hoje.

Registro, ainda, minha eterna gratidão àqueles que já não estão mais entre nós, mas que continuam vivos em minha memória e em meu coração. Tenho plena certeza de que, onde quer que estejam, estão orgulhosos desta conquista.

Expresso minha sincera gratidão à minha orientadora, que gentilmente se dispôs a me orientar durante a construção deste trabalho, com paciência, disponibilidade e comprometimento.

Por fim, não poderia deixar de agradecer às grandes amizades que construí ao longo desta jornada. Foram muitos momentos de aprendizado, risadas, conversas sérias e apoio mútuo. Tenho plena certeza de que escolhi as pessoas certas para caminharem ao meu lado, me incentivando e me motivando até aqui. A cada um de vocês, meu muito obrigado.

"O sucesso não é um destino, mas uma jornada  
de melhoria contínua." (Richard St.John, 2009)

## RESUMO

O desenvolvimento de software em muitas organizações é frequentemente marcado por desafios como falta de organização, comunicação ineficaz e dificuldade no controle das tarefas, especialmente em ambientes com recursos limitados e alta rotatividade de membros. Esses problemas impactam diretamente a eficiência e a qualidade dos projetos entregues, evidenciando a necessidade de aprimoramento dos processos internos. Diante desse cenário, este trabalho tem como objetivo principal propor melhorias no processo de desenvolvimento de software de uma empresa júnior, por meio da adoção de práticas ágeis da metodologia híbrida *Scrumban*, adaptadas à realidade da equipe estudada. Para alcançar esse objetivo, foi realizada uma análise detalhada do fluxo de trabalho vigente, identificando falhas recorrentes e pontos de atenção. Com base nesse diagnóstico, foram mapeadas e selecionadas práticas ágeis compatíveis, resultando na elaboração de um novo processo. A metodologia utilizada compreendeu a implementação de práticas como refinamento do *backlog*, planejamento com priorização dinâmica, reuniões diárias, monitoramento de métricas (*cycle time*) e uso do quadro *Kanban* via *Trello*. A validação da proposta incluiu um questionário diagnóstico para avaliar o conhecimento dos membros sobre métodos ágeis, uma reunião explicativa e o acompanhamento de uma *Sprint* experimental, avaliando os resultados por meio de *feedback* dos participantes. Os resultados obtidos apontaram melhorias significativas na organização, comunicação, controle das tarefas e percepção dos membros sobre o trabalho em equipe, com boa aceitação do novo processo e interesse em sua continuidade. Conclui-se que a adoção de práticas ágeis simples e adaptáveis pode elevar de forma relevante a qualidade e a eficiência do desenvolvimento de software em empresas juniores, sendo o modelo proposto aplicável a outras equipes com desafios semelhantes.

**Palavras-chave:** metodologia híbrida; Scrumban; empresa júnior; práticas ágeis.

## ABSTRACT

Software development in many organizations is often challenged by issues such as lack of organization, ineffective communication, and difficulty in task management, particularly in environments with limited resources and high member turnover. These challenges directly affect the efficiency and quality of delivered projects, highlighting the need for improvements in internal processes. In this context, the main objective of this study is to propose enhancements to the software development process of a junior enterprise through the adoption of agile practices from the hybrid Scrumban methodology, tailored to the reality of the studied team. To achieve this goal, a detailed analysis of the current workflow was conducted, identifying recurrent failures and key areas for improvement. Based on this diagnosis, agile practices compatible with the team's context were mapped and selected, resulting in the design of a new development process. The methodology included the implementation of practices such as backlog refinement, dynamic planning, daily meetings, metrics monitoring (cycle time), and the use of a Kanban board via Trello. Validation involved a diagnostic questionnaire to assess members' knowledge of agile methods, an explanatory meeting, and monitoring of an experimental Sprint, with results evaluated through participant feedback. The findings indicated significant improvements in organization, communication, task management, and team members' perception of teamwork, with good acceptance and interest in continuing the new process. It is concluded that adopting simple and adaptable agile practices can significantly enhance the quality and efficiency of software development in junior enterprises, and that the proposed model is applicable to other teams facing similar challenges.

**Keywords:** hybrid methodology; Scrumban; junior company; agile practices.



## LISTA DE FIGURAS

Figura 1 – Ciclo de vida do software . . . . .	17
Figura 2 – modelo fluxo da metodologia Scrum . . . . .	21
Figura 3 – Modelo fluxo da metodologia Kanban . . . . .	22
Figura 4 – Visão da metodologia proposta trabalho relacionado . . . . .	26
Figura 5 – Procedimentos metodológicos . . . . .	29
Figura 6 – Modelo descrição de atividades . . . . .	31
Figura 7 – Modelo de processo atual . . . . .	32
Figura 8 – Proposta de melhoria no processo . . . . .	36
Figura 9 – modelo fluxo da metodologia Scrum . . . . .	37
Figura 10 – Fluxo de aplicação da abordagem otimizada . . . . .	38
Figura 11 – Perguntas para validação de conhecimento do time. . . . .	39
Figura 12 – Backlog da Sprint definido . . . . .	40
Figura 13 – Resposta da questão 1 do questionário . . . . .	48
Figura 14 – Resposta da questão 2 do questionário . . . . .	49
Figura 15 – Resposta da questão 3 do questionário . . . . .	49
Figura 16 – Tarefas planejadas x Tarefas concluídas . . . . .	51
Figura 17 – Pergunta 01 - Questionário de feedback pós aplicação do processo . . . . .	52
Figura 18 – Pergunta 02 - Questionário de feedback pós aplicação do processo . . . . .	52
Figura 19 – Pergunta 04 - Questionário de feedback pós aplicação do processo . . . . .	53
Figura 20 – Pergunta 05 - Questionário de feedback pós aplicação do processo . . . . .	53
Figura 21 – Pergunta 08 - Questionário de feedback pós aplicação do processo . . . . .	54
Figura 22 – Pergunta 09 - Questionário de feedback pós aplicação do processo . . . . .	54
Figura 23 – Pergunta 11 - Questionário de feedback pós aplicação do processo . . . . .	55
Figura 24 – Pergunta 12 - Questionário de feedback pós aplicação do processo . . . . .	55
Figura 25 – Pergunta 14 - Questionário de feedback pós aplicação do processo . . . . .	56
Figura 26 – Pergunta 15 - Questionário de feedback pós aplicação do processo . . . . .	56
Figura 27 – Pergunta 17 e 18 - Questionário de feedback pós aplicação do processo . . . . .	57

## LISTA DE TABELAS

Tabela 1 – Tabela comparativa de trabalhos . . . . .	28
Tabela 2 – Mapeamento de Problemas e Práticas do Scrumban . . . . .	33

## LISTA DE ABREVIATURAS E SIGLAS

CASE	<i>Computer-Aided Software Engineering)</i>
EJ	<i>Empresa Júnior</i>
ESI	<i>European Software Institute</i>
SEI	<i>Software Engineering Institute</i>
SOFTEX	<b><i>SOFT</i></b> ware <b><i>EX</i></b> cellence
WIP	<i>Work In Progress</i>
XP	<i>eXtreme</i> Programming

## SUMÁRIO

1	INTRODUÇÃO . . . . .	13
2	OBJETIVOS . . . . .	15
2.1	Objetivos Gerais . . . . .	15
2.2	Objetivos específicos . . . . .	15
3	FUNDAMENTAÇÃO TEÓRICA . . . . .	16
3.1	Processos de software . . . . .	16
3.2	Metodologias ágeis . . . . .	18
3.2.1	<i>Scrum</i> . . . . .	19
3.2.2	<i>Kanban</i> . . . . .	21
3.3	Scrumban . . . . .	23
4	TRABALHOS RELACIONADOS . . . . .	25
4.1	Scrumban/XP: Propostas para melhorar a eficiência da gestão de projetos ágeis no desenvolvimento de software. . . . .	25
4.2	Utilização de modelos híbridos para a definição e execução de processos de software aplicados na otimização da produtividade. . . . .	26
4.3	Estudo de caso: aplicação de uma proposta de combinação de metodologias ágeis com ênfase em gestão da inovação . . . . .	27
5	METODOLOGIA . . . . .	29
5.1	Pesquisa bibliográfica . . . . .	29
5.2	Compreensão do contexto da empresa . . . . .	30
5.2.1	<i>Entrevista com a empresa júnior</i> . . . . .	30
5.2.2	<i>Diagrama do processo atual</i> . . . . .	31
5.3	Análise do processo atual e identificação dos problemas . . . . .	32
5.4	Mapeamento de práticas do Scrumban . . . . .	33
5.5	Proposta de melhoria e metodologia otimizada . . . . .	34
5.5.1	<i>Aplicação das melhorias no processo atual</i> . . . . .	34
5.5.2	<i>Justificativa da proposta</i> . . . . .	37
5.6	Aplicação da metodologia otimizada . . . . .	38
5.6.1	<i>Verificação do conhecimento do time</i> . . . . .	38
5.6.2	<i>Reunião explicativa do processo proposto</i> . . . . .	39

5.6.3	<i>Aplicação do processo proposto</i> . . . . .	40
5.6.4	<i>Feedback para validação do processo</i> . . . . .	41
6	<b>RESULTADOS</b> . . . . .	42
6.1	<b>Resultados da entrevista com a EJ para compreensão do contexto da empresa</b> . . . . .	42
6.2	<b>Resultados da entrevista com time de desenvolvimento para mapeamento de problemas</b> . . . . .	43
6.2.0.1	<i>Processo de desenvolvimento de software não é bem definido</i> . . . . .	43
6.2.0.2	<i>Cerimônias importantes não são realizadas</i> . . . . .	43
6.2.0.3	<i>Falha na comunicação entre os membros do time</i> . . . . .	44
6.2.0.4	<i>Não é realizado um gerenciamento de cronograma para o projeto</i> . . . . .	44
6.2.0.5	<i>Não são realizadas verificações e validações durante o desenvolvimento do projeto</i> . . . . .	44
6.2.0.6	<i>Problemas com mudanças nos requisitos do projeto</i> . . . . .	45
6.2.0.7	<i>Não são registrados a quantidade de bugs e falhas encontradas</i> . . . . .	45
6.2.0.8	<i>Não é realizado tratamento de requisitos inconsistentes ou ambíguos</i> . . . . .	45
6.2.0.9	<i>Dificuldade de gerenciamento da equipe</i> . . . . .	45
6.3	<b>Resultados do mapeamento das práticas do Scrumban</b> . . . . .	45
6.3.0.1	<i>Definição de backlog e refinamento</i> . . . . .	46
6.3.0.2	<i>Reuniões diárias</i> . . . . .	46
6.3.0.3	<i>Revisão da Sprint</i> . . . . .	46
6.3.0.4	<i>Retrospectiva da Sprint</i> . . . . .	46
6.3.0.5	<i>Planejamento da Sprint</i> . . . . .	47
6.3.0.6	<i>Visualização do fluxo de trabalho</i> . . . . .	47
6.3.0.7	<i>Revisão contínua do fluxo de trabalho</i> . . . . .	47
6.3.0.8	<i>Monitoramento do tempo de ciclo (Cycle Time)</i> . . . . .	47
6.3.0.9	<i>Priorização dinâmica no quadro kanban</i> . . . . .	47
6.3.0.10	<i>Métricas de defeitos no kanban</i> . . . . .	47
6.4	<b>Resultados do questionário para validação de conhecimento do time</b> . . . . .	48
6.5	<b>Resultados da Sprint pós aplicação do processo</b> . . . . .	50
6.6	<b>Resultados do questionário de feedback pós aplicação do processo</b> . . . . .	51
6.6.1	<i>Planejamento e requisitos</i> . . . . .	51

6.6.2	<i>Execução e fluxo de trabalho</i> . . . . .	53
6.6.3	<i>Métricas e visibilidade</i> . . . . .	54
6.6.4	<i>Resultados percebidos</i> . . . . .	55
6.6.5	<i>Avaliação geral</i> . . . . .	56
6.6.6	<i>Questionamentos abertos e sugestões</i> . . . . .	57
6.7	<b>Validação dos objetivos</b> . . . . .	57
7	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	59
	<b>REFERÊNCIAS</b> . . . . .	61

## 1 INTRODUÇÃO

Devido ao processo de globalização e aos avanços crescentes das tecnologias, as demandas do mercado para o desenvolvimento de novos produtos de software crescem cada vez mais ao longo dos anos. Diversos estudos mostram que desenvolver sistemas ágeis com qualidade vai além da escolha de linguagens ou ferramentas de desenvolvimento. Portanto, é importante que as organizações estabeleçam processos bem definidos para o desenvolvimento e que haja o comprometimento de todos os membros da instituição (FUGGETTA, 2000).

Diante da importância dos processos organizacionais, muitas empresas do setor de desenvolvimento de software vêm buscando maneiras para melhorar seus processos de forma que possam garantir uma melhor qualidade do produto a ser desenvolvido, realizar entregas de forma ágil e sem atrasos, e melhorar sua estrutura organizacional (SOMMERVILLE, 2011). Devido à necessidade das empresas de alcançar tais objetivos para se manterem no mercado competitivo, muitas delas passaram a adotar metodologias ágeis, como o *Scrum*, o *eXtreme Programming* (XP), o *Kanban*, entre outras, para melhorar a qualidade dos seus processos internos.

Embora a adoção de metodologias ágeis para a melhoria de processos organizacionais resultem em avanços e eleve o desempenho das empresas, é importante destacar que a implementação de uma metodologia ágil pode não ser capaz de satisfazer todos os objetivos almejados por uma organização, devido às limitações inerentes a essas metodologias (OLIVEIRA; PEDRON, 2021). Diante desse contexto, este trabalho tem como objetivo principal apresentar melhorias em um processo de desenvolvimento de software baseado em práticas da abordagem híbrida *Scrumban*, que une boas práticas da metodologia *Scrum* e *Kanban*, visando melhorar a qualidade do processo dentro de uma empresa júnior.

Para conduzir o desenvolvimento desta pesquisa foi necessário realizar alguns procedimentos metodológicos importantes como pesquisa bibliográfica, para fundamentar as bases deste trabalho. Posteriormente, foi selecionada uma *Empresa Júnior* (EJ) especializada no desenvolvimento de *software* para realizar a exploração do processo e compreender seu contexto. A escolha da EJ foi motivada devido sua estrutura adaptável e dinâmica, composta por universitários buscando vivências práticas no desenvolvimento de software. Tal cenário favorece a aplicação de metodologias ágeis, permitindo testar e ajustar o modelo proposto com maior facilidade. Além disso, a rotatividade dos membros e a curta duração dos projetos criam problemas na organização que podem ser atenuados pelo *Scrumban*, melhorando a comunicação e a eficiência. A aplicação dessa abordagem também impacta diretamente o aprendizado dos estudantes, preparando-os

para o mercado. Por fim, os resultados obtidos podem servir de referência para outras EJs ou pequenas empresas, tornando o modelo replicável em diferentes contextos.

Ademais, foi realizado um mapeamento dos problemas encontrados no processo atual da EJ. A partir destes problemas foi realizado um levantamento das boas práticas da metodologia *Scrumban*, com o intuito de realizar melhorias que otimizem o desenvolvimento do *software*. Por fim, para validar a proposta, foi conduzido um estudo de caso com uma equipe de desenvolvimento da EJ, seguido da coleta de *feedbacks* dos participantes.

Esta pesquisa encontra-se dividida da seguinte forma: no Capítulo 1 é apresentado o trabalho para contextualizar o leitor sobre o tema que será abordado e a relevância desta pesquisa. Adiante, no Capítulo 2 são apresentados os objetivos gerais e específicos que esse estudo busca atingir. No Capítulo 3 são apresentados importantes conceitos que fundamentam toda a base deste trabalho. No Capítulo 4 estão dispostos os trabalhos que se relacionam com esta pesquisa e a comparação entre eles. No Capítulo 5 é apresentado todo o processo metodológico que foi seguido para atingir os objetivos desta pesquisa. No Capítulo 6 são apresentados todos os resultados obtidos após a realizar este trabalho. Por fim, o Capítulo 7 apresenta a conclusão da pesquisa e sugere potenciais trabalhos futuros.



## 2 OBJETIVOS

### 2.1 Objetivos Gerais

Propor melhorias no processo organizacional de uma empresa júnior por meio da aplicação de práticas da abordagem ágil híbrida *Scrumban*, visando aumentar a eficiência e a eficácia dos projetos realizados.

### 2.2 Objetivos específicos

- Avaliar o processo de desenvolvimento vigente na empresa júnior, identificando seus principais pontos fortes e fragilidades;
- Diagnosticar os problemas recorrentes que impactam a eficiência e a eficácia do processo de desenvolvimento;
- Correlacionar as dificuldades encontradas com práticas recomendadas pela metodologia ágil híbrida *Scrumban*;
- Propor melhorias fundamentadas nas práticas do *Scrumban* para otimizar o processo organizacional;
- Verificar a efetividade das melhorias propostas por meio de validação prática e coleta de *feedback* da equipe.

### 3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos que fundamentam todo o desenvolvimento desta pesquisa. Para um melhor entendimento, o conteúdo abordado a seguir está dividido em tópicos fundamentais.

#### 3.1 Processos de software

O estudo acerca de processos de software teve início na década de 1980, através de eventos e *workshops* a respeito da área. Em decorrência da importância de tal estudo a respeito do processo de desenvolvimento de software foram surgindo novos trabalhos e pesquisas a respeito, assim como a criação de instituições voltadas para essa área, como “*Software Engineering Institute*” *Software Engineering Institute* (SEI) e *European Software Institute* (ESI).

Após diversos estudos realizados na área de processos de software, muitos pesquisadores e profissionais que atuam na área de desenvolvimento compreenderam que desenvolver um sistema com boa qualidade vai além de estabelecer linguagens e ferramentas e que para atingir este objetivo é necessário o comprometimento da equipe e de como está definido o processo de desenvolvimento do software na empresa (FUGGETTA, 2000). Para estabelecer um processo de *software* com melhor qualidade, é importante que o mesmo tenha atividades bem definidas, descrevendo artefatos gerados e consumidos em cada etapa do processo, e os recursos necessários em cada parte do fluxo.

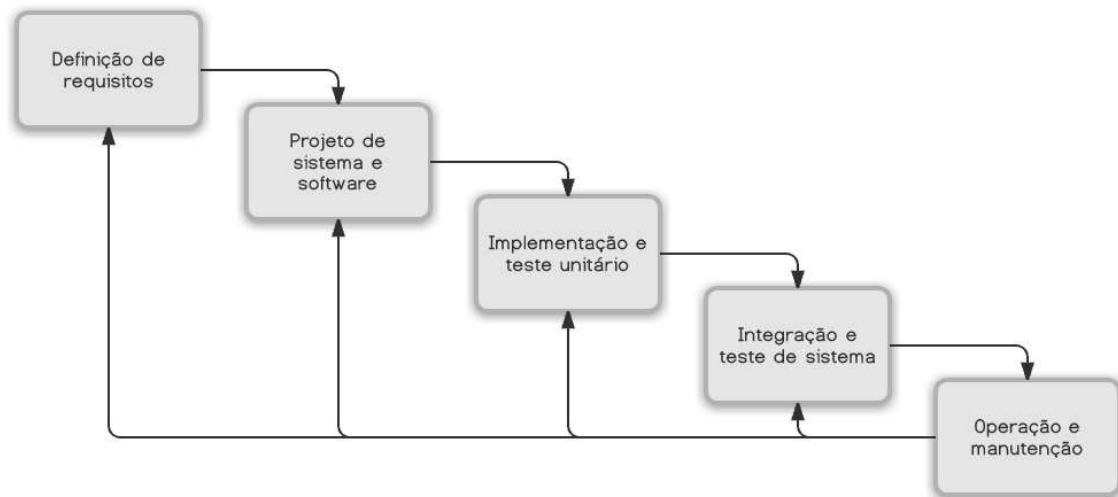
Segundo Sommerville (2011) processo de software pode ser definido como um conjunto de atividades que levam ao desenvolvimento de um produto de *software*. São considerados bastante complexos, pois não há uma estrutura a ser seguida, visto que cada organização pode estar inserida em um contexto diferente. Portanto os processos podem sofrer adaptações, o que dificulta sua automatização. Porém, Sommerville definiu em seu livro, intitulado como “Engenharia de Software”, algumas atividades essenciais que devem ser comuns a todos os diferentes processos que são:

- Especificação de requisitos do *software* a ser desenvolvido;
- Elaboração do projeto e implementação do *software* para atender aos requisitos do cliente;
- Verificação e validação do *software* que foi desenvolvido, para verificar se foi desenvolvido corretamente e se atende aos requisitos do cliente;
- Evolução do *software* para atender às novas necessidades do cliente após o sistema

desenvolvido.

Estas atividades agregam o chamado ciclo de vida do *software*, abordado na Figura 1.

Figura 1 – Ciclo de vida do software



Fonte: Sommerville (2011).

Processo de software é uma camada essencial para o desenvolvimento de sistemas, pois possibilita desenvolver sistemas de forma mais organizada, lógica e cumprindo prazos. Além disso, pode ser um pilar fundamental para o gerenciamento de projetos, pois fornece uma visão mais contextualizada e nítida na qual torna possível adotar métodos, identificar em quais atividades são gerados artefatos, em que momento são consumidos e possibilita também definir a metodologia que deve ser utilizada para realizar entregas efetivas.(PRESSMAN; MAXIM, 2021)

Ao longo dos anos, alguns modelos de processos foram desenvolvidos para atender as necessidades das empresas, estes modelos ficaram conhecidos como modelos clássicos de processo. A primeira metodologia de processo de desenvolvimento de software surgiu no ano de 1970 criado por Winston Walker Royce e foi chamada de modelo cascata, devido à sua estrutura encadeada, na qual uma atividade só pode ser iniciada após a conclusão da anterior.(ROYCE, 1987) Outra abordagem de processo é o modelo evolutivo em espiral, sua característica principal é o desenvolvimento de software seguindo uma forma em espiral em que cada volta representa uma parte do processo. Este modelo se difere do cascata pois não segue ordem sequencial. (SOMMERVILLE, 2011).

Devido as documentações extensas e a complexidade para elaborar projetos em pequenas e médias empresas surgiu a necessidade de pesquisar metodologias mais eficazes do

que os modelos tradicionais, nesse contexto surgiram as metodologias ágeis.(SOMMERVILLE, 2011)

### 3.2 Metodologias ágeis

No final da década de 1980 e o começo da década de 1990 muitas empresas tinham uma visão geral de que a melhor forma de obter o melhor *software* seria por meio de um cuidadoso planejamento de projeto garantindo a qualidade formalizada, utilizando métodos de análise, que fossem apoiados por ferramentas *Computer-Aided Software Engineering* (CASE) e mantidos por um processo rígido de desenvolvimento de *software*. Esta perspectiva ganhou força principalmente na comunidade de engenharia de *software* que visa sistemas de grande porte e longa vida. Porém, quando essas abordagens mais complexas foram adotadas no desenvolvimento de *software* por pequenas e médias empresas perceberam maior dificuldade, pois o tempo gasto com documentação era bem maior que o tempo para realizar a implementação do *software* e realização dos testes.(SOMMERVILLE, 2011)

Em fevereiro de 2001, nos Estados Unidos, cerca de 17 empresários da indústria de software se reuniram para debater sobre as metodologias, princípios e valores que devem reger o desenvolvimento de *software* com entregas rápidas e boa qualidade, a partir desta reunião foi criada a “*The Agile Alliance*” (“A aliança ágil”) com objetivo de promover os conceitos relacionados ao desenvolvimento ágil de *software* e acompanhar as organizações para que adotem esses conceitos.(FOWLER; HIGHSMITH, 2001).

A partir desta reunião surgiu o Manifesto Ágil,<sup>1</sup> que tem como propósito buscar melhores formas de desenvolver software e que segue os seguintes valores:

- Indivíduos e interações sobre processos e ferramentas;
- *Software* trabalhando sobre documentação abrangente;
- Colaboração com o cliente sobre negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Além dos valores estabelecidos para o manifesto ágil (FOWLER; HIGHSMITH, 2001) foram estabelecidos 12 princípios relacionados às metodologias ágeis que são:

- A maior prioridade é satisfazer o cliente através da entrega antecipada e contínua de produtos e programas valiosos.

<sup>1</sup> [http://andrey.hristov.com/fht-stuttgart/The\\_Agile\\_Manifesto\\_SDMagazine.pdf](http://andrey.hristov.com/fht-stuttgart/The_Agile_Manifesto_SDMagazine.pdf)

- As mudanças nos requisitos são bem vindas, mesmo no final do desenvolvimento. Processos ágeis aproveitam as mudanças como vantagem competitiva dos clientes.
- Entregar *software* funcional com frequência, de algumas semanas a alguns meses, de preferência no prazo mais curto.
- Empresários e desenvolvedores trabalham juntos diariamente durante todo o projeto.
- Construir projetos em torno de indivíduos motivados, dê-lhes o ambiente e o apoio de que necessitam e confie neles para realizar o trabalho.
- O método mais eficiente e eficaz de transmitir informações com e dentro de uma equipe de desenvolvimento é uma conversa cara a cara.
- Software funcional é a principal medida de progresso.
- Processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- Simplicidade, ou seja, a arte de maximizar a quantidade de trabalho não realizado é essencial.
- As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizadas.
- Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e, em seguida, ajusta seu comportamento em conformidade.

As metodologias ágeis constituem uma nova abordagem de metodologias de desenvolvimento de software criada para suprir a crescente demanda do mercado por processos mais ágeis, com menores ciclos de desenvolvimento e entregas rápidas.(ABRAHAMSSON, 2003). Algumas das metodologias criadas seguindo estes valores e princípios estabelecidos são *eXtreme Programming* desenvolvido por Beck nos anos 1999 e 2000, a metodologia Scrum desenvolvido por Schwaber e Beedle em 2001 e Crystal que foi desenvolvida por Cockburn em 2001.(SOMMERVILLE, 2011)

### **3.2.1 Scrum**

A metodologia ágil scrum teve sua primeira aparição em 1993, aplicada por Jeff Sutherland na Easel Corporation. No ano de 1995 essa metodologia passou por um processo de refinamento realizado por Ken Schwaber baseando-se em sua experiência com desenvolvimento de software.(BISSI, 2007). Para Bissi (2007) a metodologia Scrum define um conjunto de práticas relacionadas ao gerenciamento de um projeto que possa garantir o sucesso do mesmo. Esta metodologia busca melhorar o trabalho em equipe, a comunicação entre os membros e

maximizar a cooperação entre eles. A aplicação desta metodologia a longo prazo, poderá resultar em um aumento de produtividade.

A ideia principal do scrum é tornar o processo de desenvolvimento de software mais flexível, ou seja, fazer com que o desenvolvimento de software seja mais adepto a mudanças de requisitos entre outros pontos, visto que podem haver muitas mudanças ao longo do projeto. (SOARES, 2004)

A metodologia scrum segue um processo baseado em interações chamadas de *sprints*. Cada *sprint* apresenta um conjunto de atividades que precisam ser realizadas dentro de cada interação. Porém, antes de definir as atividades que vão acontecer na *sprint*, são realizadas reuniões para planejar como as demandas serão distribuídas no momento em que são definidas as ações e os prazos para a realização da *sprint*. Dentro deste contexto tem-se a figura do *Product Owner*, responsável por especificar o produto em nível de negócio, definir critérios de aceitação para que o desenvolvimento seja válido e definir questões relacionadas à qualidade e requisitos. Esse papel é responsável por aceitar ou rejeitar caso o produto desenvolvido não esteja de acordo com o que foi solicitado.

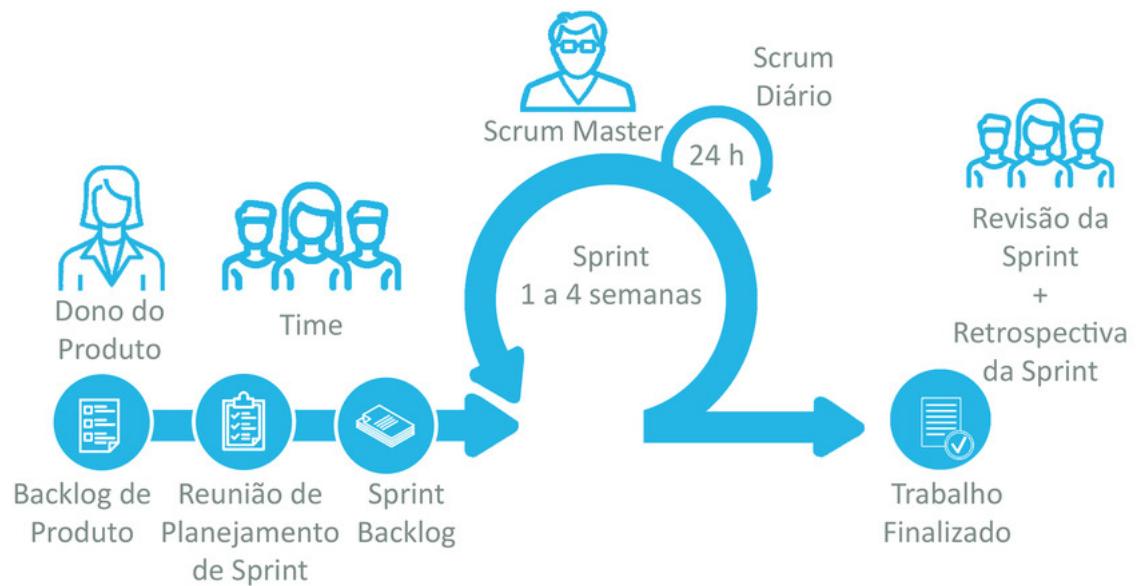
No decorrer da *sprint* são realizadas reuniões diárias rápidas (*daily meeting*), geralmente são programadas para durar em média 15 minutos. Neste tipo de reunião são abordados pelo time o que foi feito e o que será feito, e qual o impeditivo que está impactando na realização da demanda, caso exista. No final de cada *sprint* é realizada uma reunião na qual são revisados o processo e a entrega parcial desenvolvida durante a *sprint* e validam se o que foi desenvolvido segue alinhado com o que foi solicitado, esta reunião é chamada de *Sprint review*. A última etapa do ciclo scrum, se trata de uma reunião chamada de *sprint retrospective* cujo principal objetivo é sistematizar e documentar as lições aprendidas no decorrer da *sprint* e as experiências vivenciadas durante este ciclo (VASCONCELOS, 2016).

A metodologia Scrum é regida por várias práticas que contribuem bastante para a melhoria contínua do processo e para melhorar a eficiência e interação do time de desenvolvimento. O guia do Scrum <sup>2</sup> estabelece práticas fundamentais para esta metodologia. O fluxo de funcionamento da metodologia Scrum é ilustrado na Figura 2.

---

<sup>2</sup> [https://www.training.com.br/download/guia\\_do\\_scrum.pdf](https://www.training.com.br/download/guia_do_scrum.pdf)

Figura 2 – modelo fluxo da metodologia Scrum



Fonte: Nésio *et al.* (2020)

O Scrum é um processo de desenvolvimento de software interessante para ser aplicado em pequenas equipes. Em alguns estudos realizados por Coplien e Brooks, pequenas equipes de desenvolvimento que trabalham de forma independente são mais eficazes.(RISING; JANOFF, 2000).

### 3.2.2 Kanban

A abordagem Kanban foi desenvolvida durante os anos 40 dentro do sistema de produção da Toyota pelo engenheiro Taiichi Ohno. A motivação de Ohno para o desenvolvimento desta abordagem surgiu da forma como são organizadas o fluxo de estoque e prateleiras de supermercados. Após análise, Ohno resolveu aplicar uma técnica dentro do sistema de produção da Toyota, na qual buscava limitar e manter um melhor controle do *Work In Progress* (WIP) (BHAVSAR *et al.*, 2020) dentro do sistema de produção utilizando cartões e sinais visuais para equiparar os níveis de estoque com a utilização dos recursos para desenvolver os veículos. Tal processo de fabricação utilizado pelo sistema de produção Toyota ficou conhecido como *Just in Time*.(ANDERSON; CARMICHAEL, 2016)

A abordagem Kanban é uma técnica mais enxuta para o desenvolvimento ágil de software, a mesma possibilita uma melhor visualização do fluxo de produção, pois mostra todas as etapas do processo a ser seguido pela equipe de desenvolvimento.

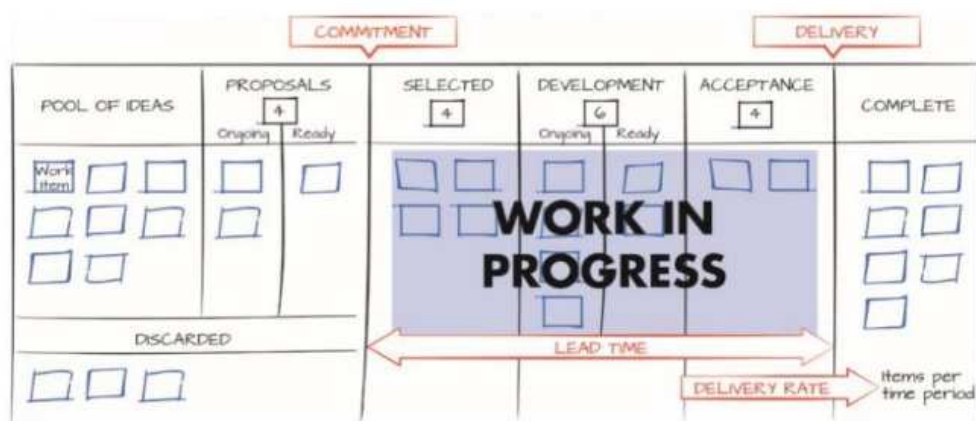
Um dos benefícios desta abordagem é a fácil identificação de falhas e “gargalos”,

que se trata de um congestionamento de atividades em um fluxo de desenvolvimento, ou mesmo desperdícios de recursos que podem ocorrer ao longo do processo, tudo isso devido à visão clara do andamento do processo que esta abordagem fornece.(LIKER, 2004)

O fluxo de desenvolvimento na abordagem *Kanban* se inicia após a definição dos requisitos do produto que será desenvolvido onde serão transformados em uma lista de itens e em seguida priorizados. Após a etapa de definição e priorização, estes itens entram para uma esteira de desenvolvimento, na qual, seguindo o limite estabelecido pelo WIP cada funcionário recebe apenas um item para desenvolver. Esta esteira é dividida em três categorias que representam o estado em que o item pode se encontrar no decorrer do desenvolvimento que são: “A fazer”, “Fazendo” e “Feito” (BHAVSAR *et al.*, 2020). A Figura 3 mostra o quadro Kanban, que organiza as tarefas em etapas do processo e limita o trabalho em andamento para evitar sobrecarga. As setas na base indicam o tempo de ciclo e a taxa de entrega, auxiliando no acompanhamento do fluxo de trabalho.

No livro intitulado como *Kanban Essencial Condensado* (ANDERSON; CARMICHAEL, 2016) apresenta práticas que são fundamentais e sustentam os princípios desta metodologia.

Figura 3 – Modelo fluxo da metodologia Kanban



Fonte: Anderson e Carmichael (2016).

De acordo com (ANDERSON; CARMICHAEL, 2016) a metodologia Kanban é guiada por alguns valores que envolvem as seguintes questões:

- Transparência: consiste em tornar o fluxo de trabalho mais visível, incluindo suas restrições;
- Equilíbrio: consiste na compreensão entre diferentes aspectos e pontos de vista;



- Colaboração: está relacionado ao comprometimento da equipe;
- Foco no cliente: consiste em explorar mais a perspectiva do cliente do que perspectivas de atividades ou produtos;
- Fluxo: ao chegar nesse estado as pessoas conseguem ser mais produtivas;
- Liderança: uma boa gestão é fundamental para um melhor condicionamento de prosperidade e gerar uma melhor auto organização da equipe;
- Entendimento: é importante que as pessoas tenham conhecimento do processo;
- Acordo: é necessário estabelecer um acordo que motive as pessoas para garantir um melhor desempenho e sustentável;
- Respeito: é necessário respeitar a forma de trabalho bem como os papéis definidos.

### 3.3 Scrumban

A abordagem Scrumban é uma metodologia híbrida surgida a partir da junção da metodologia Scrum e Kanban, esta foi introduzida ao mundo pela primeira vez por Corey Ladas em 2009, com seu livro *“Essays on Kanban Systems for Lean Software Development”*. Em seu livro, Ladas (2009) descreve a abordagem Scrumban como uma transição da abordagem de desenvolvimento Scrum para uma abordagem de desenvolvimento mais evoluída

A utilização desta abordagem surgiu a partir da necessidade de um melhor gerenciamento da abordagem Scrum, que por mais que seja uma abordagem relativamente simples, em um primeiro momento ao tentar aplicá-la corretamente seguindo as cerimônias necessárias para melhorar questões de agilidade e desempenho, pode se tornar um pouco mais complexa. Então muitas organizações, para manter uma certa facilidade na adoção da abordagem acabam ocultando ou alterando partes principais da metodologia, como as cerimônias, ignorando a importância de se compreender os valores e princípios básicos que regem a metodologia.(REDDY, 2015)

Segundo Khan (2014), a abordagem Scrumban se utiliza das melhores práticas e características dos dois modelos, utiliza normas prescritas na abordagem Scrum, como compartilhamento de informações, a capacidade dos times se auto organizarem e da abordagem Kanban se utilizar dos incentivos para melhoria do processo, identificando gargalos e falhas quanto ao processo.

De acordo com Reddy (2015) adoção da metodologia Scrumban pode chegar a alguns resultados como:

- Permitir uma maior autonomia para o time, ou seja, que as equipes de desenvolvimento ou até mesmo as organizações sejam capazes de gerenciar as cerimônias do Scrum.
- Mostrar ferramentas e contribuir para o desenvolvimento de habilidades para lidar com problemas relacionados à implementação da metodologia Scrum ou até mesmo outras metodologias ágeis.
- A abordagem é bastante flexível para que seja possível aplicá-la da melhor forma para ter um alto nível de desempenho.
- Está estruturado de uma forma que apoia tanto a evolução quanto a revolução e suporta todos os níveis de aprendizagem e compreensão.

Scrum e Kanban são metodologias ágeis amplamente utilizadas no desenvolvimento de software. O Scrum estrutura o trabalho em ciclos chamados sprints, com duração fixa e foco em entregas incrementais, além de definir papéis, cerimônias regulares e planejamento antecipado das tarefas (SCHWABER; SUTHERLAND, 2011). Por outro lado, o Kanban é uma metodologia visual de gestão de fluxo contínuo, que não exige divisão do trabalho em períodos determinados, mas sim prioriza a flexibilidade, a limitação do trabalho em andamento (WIP) e o acompanhamento contínuo das tarefas (ANDERSON, 2010). Enquanto Scrum enfatiza organização por ciclos e rituais, Kanban possibilita adaptação constante ao processo e controle do fluxo, sendo ambas metodologias que podem ser combinadas no Scrumban para potencializar a eficiência das equipes.

A metodologia híbrida Scrumban apresenta uma transição suave entre as abordagens Scrum e Kanban que proporcionam uma maior flexibilidade na qual permite que as equipes explorem e adotem de maneira gradual as práticas mais alinhadas com suas necessidades específicas, mantendo uma estrutura que apoia a entrega iterativa e colaborativa. Portanto a utilização desta abordagem para desenvolvimento deste trabalho, é pautada na capacidade de combinar elementos do Scrum e do Kanban oferecendo a adaptação necessária para lidar com as demandas variáveis do processo de desenvolvimento de software e ao mesmo tempo, promover uma cultura de melhoria contínua (ALQUDAH; RAZALI, 2018). A metodologia Scrumban se utiliza das cerimônias do Scrum, no caso, reuniões de planejamento, diárias e retrospectiva, entre outras práticas como definição de backlog e priorização de itens. Além disso, para se manter um melhor gerenciamento e organização do projeto, são adotados algumas práticas do Kanban como os painéis para visualização e o WIP. (REIS, 2021).

## 4 TRABALHOS RELACIONADOS

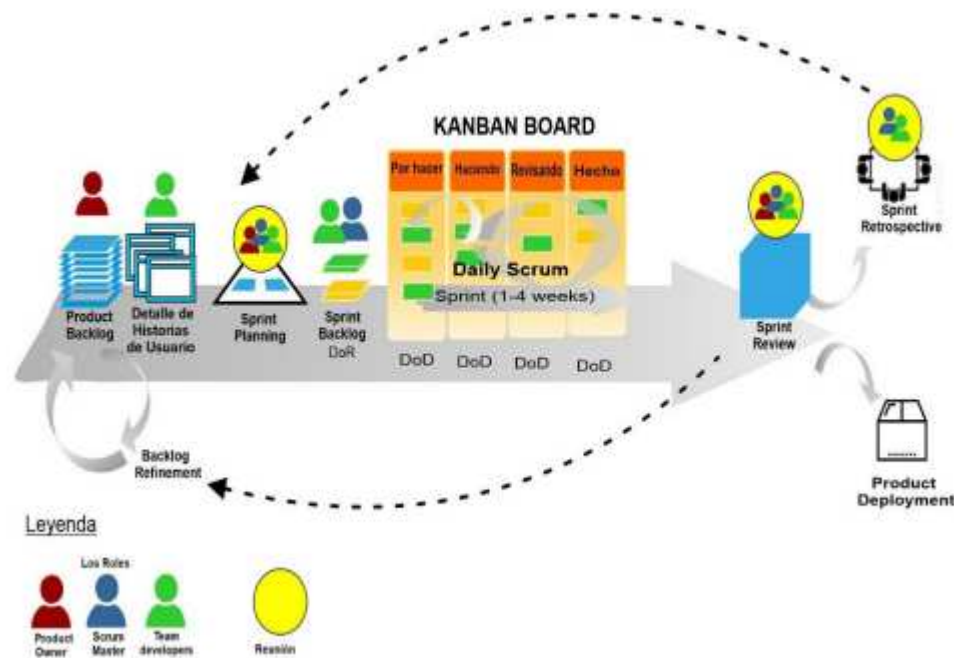
Os trabalhos relacionados que foram encontrados se tratam de trabalhos que envolvem a elaboração de melhorias no processo de desenvolvimento de software baseado em boas práticas das metodologias ágeis, realizando uma combinação entre as mesmas para se obter as melhores práticas de cada metodologia utilizada. A seguir são apresentados os trabalhos detalhadamente.

Para realizar a busca bibliográfica, foram utilizadas palavras-chave como "melhoria de processo", "práticas ágeis", "metodologia híbrida" e "metodologia ágil". Essas palavras foram aplicadas em revistas científicas, repositórios acadêmicos e periódicos especializados, com o objetivo de localizar artigos e trabalhos relevantes que se relacionam com esta pesquisa.

### 4.1 Scrumban/XP: Propostas para melhorar a eficiência da gestão de projetos ágeis no desenvolvimento de software.

Em seu trabalho Patilla, Alviar e Conislla(2023) propõem uma metodologia híbrida envolvendo as metodologias ágeis *Scrum*, *Kanban* e *Extreme Programming* (XP) reunindo aspectos positivos dessas metodologias para se obter uma metodologia híbrida que possa atingir os objetivos da pesquisa como adotar estratégias de desenvolvimento incremental, gerar conhecimentos nas equipes, auto-organização por meio de iterações, priorização de tarefas e gerar qualidade no resultado. No processo metodológico do trabalho em questão foram necessários realizar análises comparativas de forma detalhada, para buscar as fragilidades e potencialidades que caracterizam cada metodologia. Primeiramente foram comparadas as metodologias Scrumban e XP analisando diferentes aspectos de cada uma e em seguida foi realizada uma análise de integração para reforçar as fragilidades encontradas e a partir disso elaborar uma metodologia que melhore a gestão de projetos ágeis para aplicar em uma empresa de desenvolvimento. A Figura 4 apresenta o fluxo de trabalho ágil, combinando Scrum e Kanban. As tarefas começam no backlog, passam pelo planejamento da sprint e são organizadas no quadro Kanban em etapas (“Por fazer”, “Fazendo”, “Revisando”, “Feito”). O time realiza reuniões diárias, revisa o progresso, faz a retrospectiva e, ao final, entrega o produto. Os ícones representam os papéis da equipe e os momentos de reunião.

Figura 4 – Visão da metodologia proposta trabalho relacionado



Fonte: Patilla *et al.* (2023)

A abordagem proposta atingiu resultados significativos, melhorando aspectos de comunicação entre as equipes, por meio de interações contínuas, aprimorando as práticas direcionadas a gestão de projetos de software. Além disso, propiciou uma melhor dinamicidade dos processos de planejamento e execução.

A relação deste trabalho com a pesquisa se estabelece a partir da utilização de práticas ágeis das metodologias *Scrum*, *Kanban* e *XP* para elaboração de um novo processo otimizado visando resolver problemas internos da organização e garantir uma melhor qualidade do resultado.

#### 4.2 Utilização de modelos híbridos para a definição e execução de processos de software aplicados na otimização da produtividade.

O trabalho de Martinhago (2018) apresenta um modelo híbrido de processo de desenvolvimento de software elaborado no intuito de atingir os seguintes objetivos: a) Otimizar a performance do time de desenvolvimento; e b) Eliminar gargalos no processo de elicitação de requisitos, desenvolvimento e testes. Na metodologia do trabalho abordado primeiramente foram realizados levantamentos bibliográficos para formar bases de conhecimento a respeito de processos de desenvolvimento, em seguida foram definidas métricas relacionadas a desempenho

para que fosse possível mensurar o desempenho obtido e após isso, foi realizada análise do processo de desenvolvimento já existente na empresa para identificar "gargalos" no processo e com base nesses problemas encontrados desenvolver um modelo híbrido baseado nas metodologias tradicionais como cascata e espiral e em metodologias ágeis como *Scrum* e *Extreme Programming*. Após a realização da aplicação foram feitas análises das métricas de desempenho e foi constatado que a metodologia híbrida atingiu melhor eficiência no tempo de trabalho durante os sprints, reduzindo o tempo de ociosidade e eliminando gargalos no processo de desenvolvimento.

Com a utilização de práticas ágeis e tradicionais para resolver problemas interno no desenvolvimento de software, pode-se estabelecer uma correlação com esta pesquisa que também busca resolver problemáticas no processo através da utilização de práticas de outras metodologias, porém limitado às metodologias ágeis.

#### **4.3 Estudo de caso: aplicação de uma proposta de combinação de metodologias ágeis com ênfase em gestão da inovação**

O trabalho desenvolvido por Dias (2022) com o título “Estudo de caso: aplicação de uma proposta de combinação de metodologias ágeis com ênfase em gestão da inovação” apresenta um estudo de caso realizado a partir da proposta de aplicação de uma metodologia híbrida, reunindo boas práticas de algumas metodologias como Scrum, Kanban, XP, Lean, entre outras. O objetivo da pesquisa consiste em propor uma metodologia ágil baseada na combinação de outras metodologias ágeis aliadas à gestão de inovação. Este estudo de caso foi realizado dentro de uma startup. Na metodologia deste trabalho foram realizadas pesquisas bibliográficas, análises de trabalhos relacionados e feito comparativos com casos reais, em seguida foi aplicado um questionário com o time para entender o nível de experiência em projetos de software. Após a realização dessas etapas foi realizado o estudo de caso dentro da startup, avaliando o modelo proposto e uma entrevista com os participantes para validar a evolução da equipe. Com o resultado obtido do estudo de caso realizado foi possível observar que é de suma importância a utilização de metodologias ágeis devido a sua flexibilidade e fácil adaptação a cada projeto. Segundo a autora, devido a falta de experiência do time, as entregas de atividades planejadas foram impactadas de forma negativa. Outro aspecto importante foi a falta de comunicação do time, que não foi ocasionado por conta do modelo, e sim devido a falta de experiência. Na Tabela 1 é realizado um comparativo entre os trabalhos relacionados e esta pesquisa.

A metodologia híbrida aplicada neste trabalho foi construída com base em práticas

de outras metodologias ágeis e obteve um resultado importante, porém com algumas limitações. Portanto, uma relação pode ser estabelecida com essa pesquisa que também busca resolver problemas internos através da melhoria de processos pela utilização de metodologia híbrida.

Tabela 1 – Tabela comparativa de trabalhos

	(DIAS, 2022)	(MARTINHAGO, 2018)	(PATILLA <i>et al.</i> , 2023)	Este Trabalho
Tem como alvo pequenas empresas	Sim	Não	Não	Sim
Baseia-se apenas em metodologias ágeis	Sim	Não	Sim	Sim
Realiza estudo de caso	Sim	Sim	Sim	Sim
Propõe melhorias no processo	Não	Não	Não	Sim
Mensura resultados por meio de feedbacks	Sim	Não	Sim	Sim

Fonte: pelo autor.

## 5 METODOLOGIA

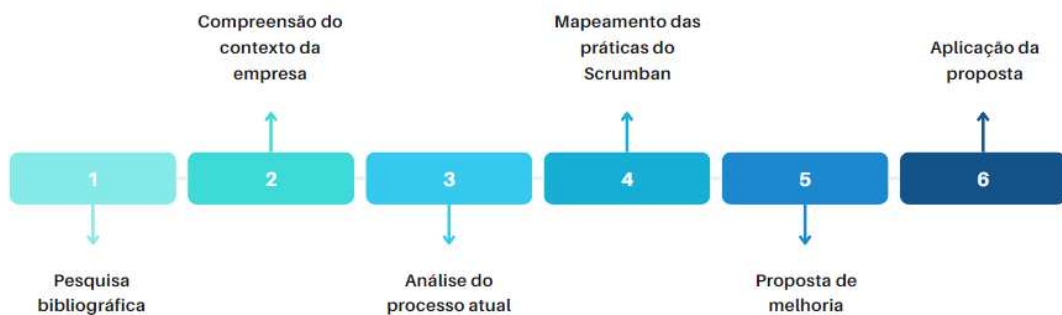
A pesquisa realizada é de natureza aplicada, com abordagem qualitativa, e caracteriza-se como um estudo de caso, uma vez que foi conduzida diretamente no ambiente da empresa júnior, analisando o processo real e testando a aplicação das melhorias propostas.

Além disso, todos os participantes deste trabalho participaram de forma voluntária e foram devidamente informados sobre os objetivos do estudo. Foi garantido sigilo e confidencialidade das informações coletadas, sendo que todos assinaram o Termo de Consentimento Livre e Esclarecido (TCLE) que encontra-se no Apêndice B.

Para a realização desta pesquisa foram estabelecidos alguns procedimentos metodológicos para assim atingir o objetivo proposto.

Contudo, esta pesquisa é constituída pelas seguintes etapas: Pesquisa bibliográfica, Compreensão do contexto da empresa, Análise do processo atual, Mapeamento de práticas do Scrumban, Proposta de melhoria e Aplicação da metodologia. Na Figura 5 é apresentada cada etapa da metodologia.

Figura 5 – Procedimentos metodológicos



Fonte: pelo autor.

### 5.1 Pesquisa bibliográfica

A primeira etapa do estudo realizado consistiu em realizar pesquisas para consolidar as bases de conhecimento deste trabalho no período de julho a agosto de 2023, reunindo conceitos e informações essenciais para se desenvolver a fundamentação teórica. Além disso, foram buscados por trabalhos relacionados e analisados aspectos como as metodologias utilizadas, as abordagens e os resultados obtidos, que foram publicados dentro de um período de 18 anos, na qual foram selecionados aproximadamente 28 trabalhos. Para esta pesquisa foram utilizadas

palavras-chave, como: otimização de processo, metodologias ágeis, processos de software, metodologias híbridas, Scrumban e otimização de processos. As buscas foram realizadas em algumas bases de conhecimentos importantes como IEEE, Periódicos da Capes e *Google Scholar*.

Contudo, após a definição de palavras-chaves e das bases para realizar a busca, foram realizadas análises de artigos, livros e teses relacionados ao tema desta pesquisa para uma melhor compreensão sobre este trabalho. Os trabalhos selecionados pelo potencial de contribuição para a fundamentação desta pesquisa encontram-se dispostos no Capítulo 4.

## **5.2 Compreensão do contexto da empresa**

A segunda etapa desta pesquisa consistiu em reunir informações a respeito da EJ, para a qual será proposto um modelo de melhoria de processo, analisando seu contexto e seu processo de desenvolvimento. A escolha da empresa júnior está relacionada ao fato de sua estrutura ser adaptável e dinâmica, composta por universitários buscam viver experiências práticas no desenvolvimento de software. Tal cenário favorece a aplicação de metodologias ágeis, permitindo testar e ajustar o modelo proposto com maior facilidade. Além disso, o pesquisador já participou da mesma como membro e diretor de projetos, buscando experiências para o mercado de trabalho. Durante este período, foi possível identificar problemas internos que levaram a atrasos na entrega de projetos e foi possível identificar também que muitas práticas da engenharia de software não são aplicadas dentro da EJ, o que impossibilita o estudante de engenharia de software obter experiências práticas voltadas para sua área de formação. Uma das motivações para a realização desta pesquisa é despertar uma cultura de boas práticas da engenharia de software e induzir a organização a trabalhar na melhoria de seus processos organizacionais.

Contudo, para a realização da exploração do processo, foi necessário dividir esta seção em algumas etapas bastante significativas que consistem em: realização de entrevista, sintetização de informações, elaboração do processo e validação do processo com a EJ.

### **5.2.1 Entrevista com a empresa júnior**

Foi realizada uma entrevista com a presidente da EJ, de maneira remota devido a disponibilidade de ambas as partes, para obter mais informações sobre a organização com a finalidade de compreender o contexto no qual a empresa está inserida, a quantidade de membros, as dificuldades enfrentadas pelo time, como é feito o gerenciamento dos projetos e das




equipes. Além disso, foram obtidas mais informações a respeito do processo utilizado pela EJ no desenvolvimento dos projetos.

Para a realização desta entrevista elaboraram-se perguntas abertas para se obter um melhor detalhamento das informações. No início da entrevista, foram realizadas algumas perguntas introdutórias a respeito da estrutura organizacional, hierarquia e quantidade de membros. Em seguida, deu-se início às perguntas voltadas para o processo de desenvolvimento utilizado, e sobre os desafios enfrentados no desenvolvimento de software pela EJ. No Apêndice A é apresentado o roteiro da entrevista realizada.

### 5.2.2 Diagrama do processo atual

Após a realização da entrevista, foi elaborado um diagrama de atividades baseando-se na descrição das informações fornecidas. Primeiramente foram analisadas e sintetizadas as informações referentes ao fluxo seguido durante a etapa de desenvolvimento de software, em seguida foi realizada a descrição de cada atividade identificando quais artefatos eram requeridos e quais eram gerados, bem como identificar os participantes e responsáveis pela atividade. Na Figura 6 é apresentado um exemplo da descrição que foi realizada de cada atividade.

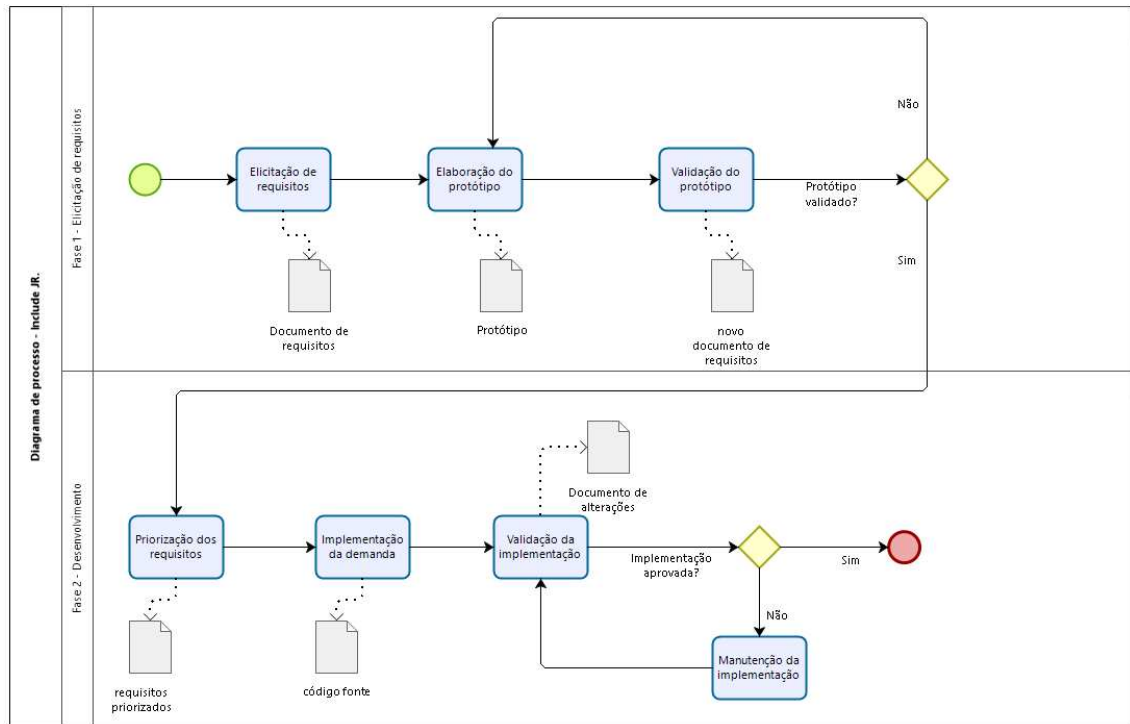
Figura 6 – Modelo descrição de atividades

Atividade:	Elaboração do protótipo
Descrição:	É criado um protótipo inicial a partir dos requisitos levantados.
Critério de Entrada:	Realização da elicitação de requisitos 
Responsáveis:	Designer
Participantes:	Designer
Pré-atividade:	Elicitação de requisitos e Validação do protótipo
Artefatos Requeridos:	versão inicial de requisitos
Artefatos Gerados:	Modelo de protótipo do software
Ferramentas:	FIGMA

Fonte: pelo autor.

Após a elaboração do diagrama do processo desenvolvido, foi realizada uma validação com o diretor de projetos e 4 membros do time de desenvolvimento para verificar se o diagrama está alinhado com o processo utilizado e assim partir para a etapa seguinte, que se trata de identificar problemas no processo. Na Figura 7 é apresentado o modelo de processo atual da EJ.

Figura 7 – Modelo de processo atual



Fonte: pelo autor.

### 5.3 Análise do processo atual e identificação dos problemas

Após a elaboração do processo atual, visando obter informações mais precisas e que ajudem na identificação de problemas, foi realizada outra entrevista, porém, com o atual diretor de projetos da empresa júnior, juntamente com 4 membros do time de desenvolvimento. Durante a entrevista, foram levantados questionamentos sobre o processo de desenvolvimento de software que é seguido atualmente e os desafios enfrentados pela EJ. Para a entrevista, foram realizadas perguntas abertas, visando um detalhamento melhor das informações, iniciando com alguns questionamentos introdutórios e partindo para perguntas mais específicas sobre o processo para identificar os problemas.

Após uma análise das informações obtidas a partir da entrevista, foram identificados

e mapeados problemas que impactam diretamente o desenvolvimento do software, ocasionando o atraso do projeto.

#### 5.4 Mapeamento de práticas do Scrumban

Após o levantamento dos problemas no processo de desenvolvimento da organização, foi realizado um mapeamento das práticas do Scrumban que poderiam mitigar esses desafios. Ademais, para a seleção dessas práticas em artigos, revistas e periódicos, foi utilizado como base os problemas identificados pela EJ durante a entrevista. Na Tabela 2 apresentada a seguir, estão descritas as práticas mapeadas de cada metodologia e a respectiva problemática que visa resolver.

Tabela 2 – Mapeamento de Problemas e Práticas do Scrumban

<b>Problema Identificado</b>	<b>Prática do Scrum</b>	<b>Prática do Kanban</b>	<b>Justificativa</b>
Processo de desenvolvimento de software não é bem definido	Backlog do Produto e Refinamento	Visualização do fluxo de trabalho	Permite um processo estruturado com backlog priorizado e um fluxo de trabalho visível para toda a equipe.
Cerimônias importantes não são realizadas	Reuniões diárias, Revisão e Retrospectiva da Sprint	Revisão de fluxo de trabalho	Melhora a comunicação da equipe e garante um processo de melhoria contínua.
Falha na comunicação entre os membros do time	Reuniões diárias	Uso de quadros Kanban	Facilita o alinhamento da equipe e a visibilidade do progresso das tarefas.
Não é realizado um gerenciamento de cronograma	Planejamento da Sprint	Monitoramento do tempo de ciclo	Permite um planejamento prévio e um acompanhamento do tempo médio de desenvolvimento de cada item.
Não são realizadas verificações e validações	Revisão da Sprint	Inspeção contínua do fluxo	Garante que o código seja revisado regularmente, evitando falhas e retrabalho.
Problemas com mudanças nos requisitos	Refinamento do Backlog	Priorização dinâmica no quadro Kanban	Permite ajustes nos requisitos de forma controlada e transparente.
Não são registrados bugs e falhas	Revisão da Sprint	Métricas de defeitos no Kanban	Mantém um histórico de falhas e melhorias para otimizar a qualidade do software.
Requisitos inconsistentes ou ambíguos	Refinamento do Backlog	Revisão contínua dos itens no quadro	Garante que os requisitos estejam bem definidos antes do desenvolvimento.
Dificuldade de gerenciamento da equipe	Reuniões diárias, Sprint Planning	Visualização do fluxo de tarefas	Facilita o acompanhamento das atividades e a distribuição das demandas.

## 5.5 Proposta de melhoria e metodologia otimizada

Com base na análise do processo atual e no mapeamento das práticas do Scrumban (seção 5.4), esta seção propõe melhorias específicas para otimizar o fluxo de desenvolvimento de software na empresa Júnior. As melhorias foram estruturadas de forma alinhada às práticas híbridas, combinando o foco no fluxo do Kanban com as cerimônias e papéis do Scrum (AHMAD *et al.*, 2013).

### 5.5.1 Aplicação das melhorias no processo atual

Para aprimorar o fluxo de trabalho, foram realizadas as seguintes mudanças de forma estratégica no processo:

**Denominação de papéis:** como a EJ trabalha com equipes pequenas de até 5 pessoas, então foi realizada uma adaptação de papéis definidos pelo Scrum (SCHWABER; SUTHERLAND, 2011). Portanto, foi realizada a seguinte distribuição dentro do processo:

- **Analista de requisitos:** responsável por elicitar os requisitos, criar o backlog do produto e realizar o refinamento dos mesmos. Além disso, deverá participar das reuniões de planejamento da Sprint para alinhar com o time as prioridades advindas por parte do cliente.
- **Líder técnico:** responsável por realizar as cerimônias estabelecidas no processo, buscar resolução dos problemas e validar os itens implementados na etapa de review.
- **Desenvolvedor:** responsável por realizar o desenvolvimento das demandas priorizadas de acordo com o que foi solicitado e reportar os impedimentos encontrados para o líder técnico.
- **Analista de qualidade:** responsável por realizar os testes da implementação desenvolvida, encontrar, reportar e documentar os bugs para serem corrigidos.

**Definição do Backlog do produto:** o contato inicial com o cliente é essencial para o entendimento do produto e para garantir o alinhamento de todos quanto às funcionalidades a serem desenvolvidas (ALVES, 2022). Na fase atual de elicitação de requisitos, observa-se que o backlog gerado não apresenta um bom detalhamento e requer uma análise mais criteriosa quanto à completude e ambiguidade. Para isso, foi incorporada a seguinte prática:

- **Refinamento de backlog:** o analista de requisitos elabora uma lista contendo as funcionalidades priorizadas e descrevendo-as de forma detalhada avaliando clareza, completude,

ambiguidade e consistência permitindo que o time de desenvolvimento trabalhe de forma mais assertiva e ágil.

**Planejamento da Sprint:** atualmente, a equipe trabalha com sprints semanais para desenvolver e entregar itens que agreguem valor ao cliente. Portanto, para garantir o alinhamento do time e a correta definição das prioridades, foi adotada uma reunião de planejamento no início de cada sprint. Nessa reunião, os itens do *backlog* refinado são avaliados e priorizados, garantindo que todos compreendam os objetivos da sprint e estejam comprometidos com suas entregas. Essa prática contribui para reduzir falhas de comunicação e minimizar problemas relacionados à priorização das tarefas. Além disso, com a realização do planejamento foi adotada a prática de priorização dinâmica no quadro, permitindo uma maior flexibilidade nas mudanças de requisitos (SCHWABER; SUTHERLAND, 2011).

- **Priorização dinâmica do quadro kanban:** após a realização do planejamento da Sprint, em caso de necessidade, as demandas priorizadas podem ser reorganizadas para atender as novas necessidades das partes interessadas.

**Visualização do fluxo de trabalho:** o quadro kanban ajuda na visualização do trabalho que está em andamento, bem como deixa explícito o processo de desenvolvimento para toda a equipe (LIKER; HOSEUS, 2010). No quadro do projeto é separado por colunas com as seguintes divisões: Backlog, Sprint Backlog, A fazer, Fazendo, Impedimento, Revisão, Teste, Concluído. Logo as demandas que foram priorizadas e mapeadas individualmente são representadas por cartões que seguem o fluxo do quadro kanban até finalizarem na etapa de concluída. Na criação do quadro do projeto visando melhorar a qualidade do desenvolvimento e evitar retrabalho, foi inserida a seguinte prática no processo:

- **Inspeção contínua do fluxo:** o líder do projeto realiza uma revisão do item desenvolvido antes de chegar na etapa de teste para avaliar a lógica implementada, padrões de implementação, além de identificar potenciais bugs para serem corrigidos antes de seguir para as próximas etapas.

**Reuniões diárias, revisão e retrospectiva:** essas práticas contribuem para estabelecer uma melhor comunicação e agilidade na resolução de problemas durante o desenvolvimento das demandas (VASCONCELOS, 2016). Após o início da Sprint, serão realizadas reuniões diárias com duração máxima de 15 minutos para alinhar o andamento das demandas e resolver problemas. Ao finalizar a Sprint será realizada uma reunião para validar todas as entregas da sprint e passar um feedback em relação ao que foi feito. Além disso, é realizada uma reunião de

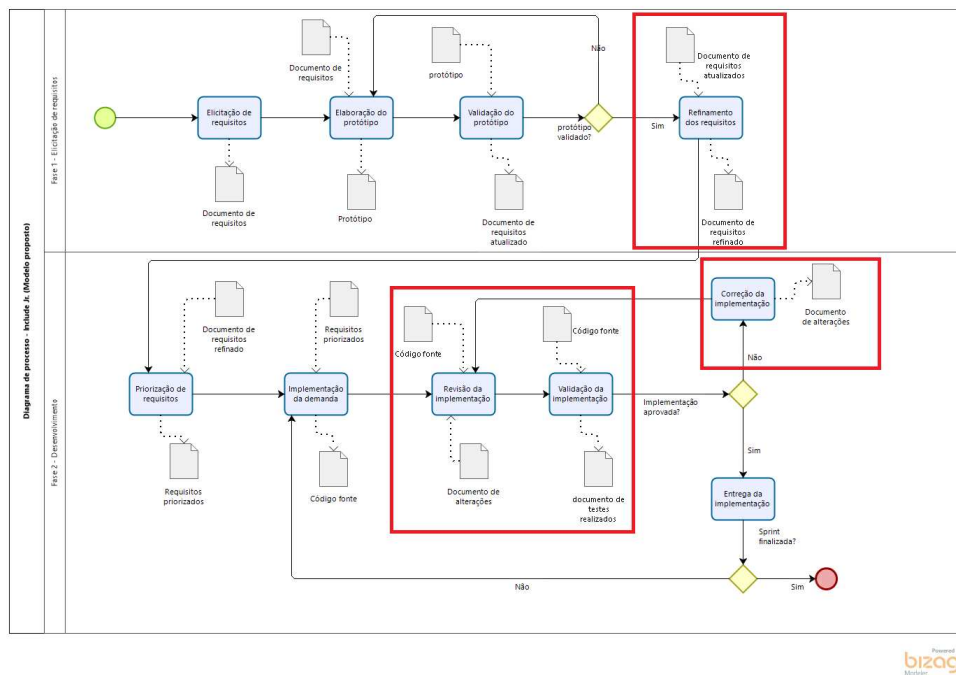
retrospectiva da Sprint para apresentar pontos positivos da Sprint e pontos de melhoria para a próxima Sprint.

**Monitoramento de tempo de ciclo:** o *cycle time* é uma métrica importante para monitorar o tempo que cada item permanece na esteira, contribuindo na identificação de gargalos e uma melhor estratégia para orçamento de horas para as demandas. Na fase de desenvolvimento adotou-se a utilização dessa prática para a criação de métricas, quando o item tiver seu desenvolvimento iniciado o tempo de ciclo estará sendo monitorado até a entrega do item (AGRAWAL; CHARI, 2007).

**Métricas de defeitos no kanban:** manter os registros de defeitos encontrados durante a etapa de desenvolvimento é essencial para medir a qualidade do processo e do produto. Na fase de desenvolvimento, em uma das colunas do quadro kanban, será mantido um histórico de problemas encontrados na Sprint, relatando o problema que ocorreu, o cenário que gerou a falha e a criticidade da falha.

Para obter uma melhor visualização e compreensão da proposta na Figura 8 é apresentado o diagrama do processo após serem adicionadas as melhorias abordadas no tópico 5.5.1

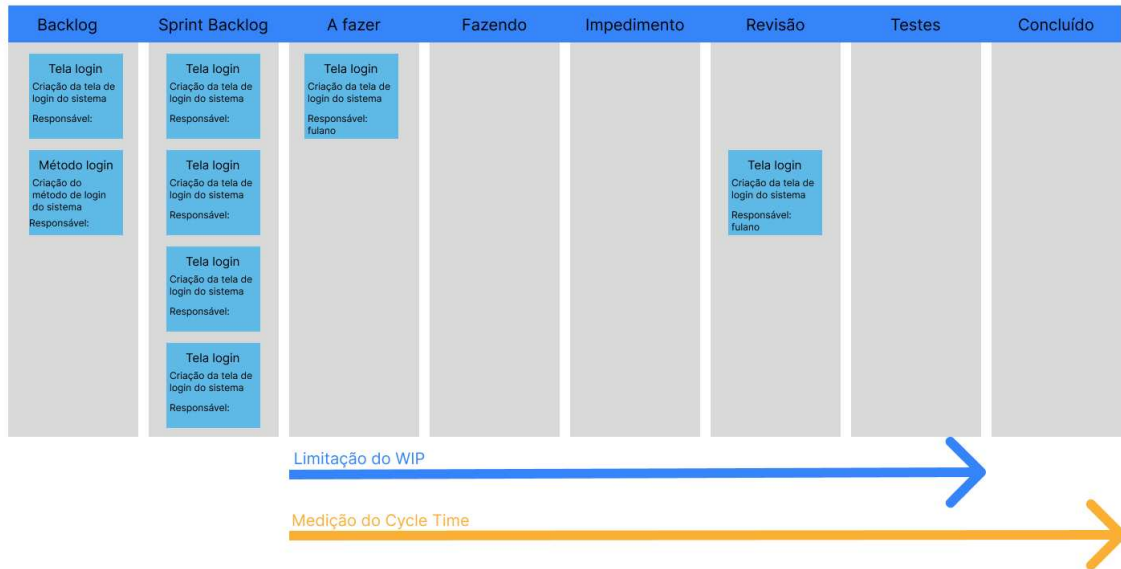
Figura 8 – Proposta de melhoria no processo



Fonte: pelo autor.

Na Figura 9 é apresentada a forma como foi visualizado o processo pelo time de desenvolvimento e como será realizada a medição do tempo de ciclo de cada atividade e garantir que o WIP esteja sendo seguido.

Figura 9 – modelo fluxo da metodologia Scrum



Fonte: pelo autor.

### 5.5.2 Justificativa da proposta

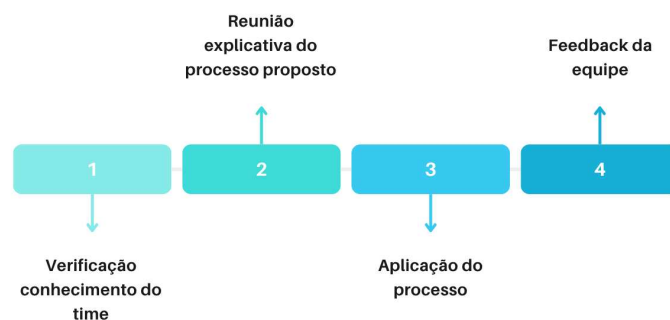
As alterações realizadas no processo atual foram fundamentadas nas problemáticas identificadas na seção 5.3.1, sendo diretamente relacionadas às práticas mais adequadas da metodologia Scrumban, conforme o mapeamento apresentado na seção 5.4.1. A escolha dessas práticas levou em consideração o contexto da empresa júnior, que demanda flexibilidade, organização visual e uma rotina de trabalho eficiente.

A proposta de melhoria visa sanar as principais falhas observadas no processo atual, tais como: baixa clareza nos requisitos, falhas de comunicação, dificuldade em adaptar-se a mudanças, ausência de indicadores de qualidade e de métricas de produtividade. A combinação de práticas do Scrum e do Kanban se mostrou ideal para esse cenário, permitindo a estruturação de ciclos curtos de entrega com um fluxo contínuo de trabalho e controle visual. Na Tabela 2 está apresentada de forma sucinta o mapeamento e a combinação das práticas que visam resolver os problemas e a justificativa.

## 5.6 Aplicação da metodologia otimizada

Com base nas melhorias propostas para o processo atual (Seção 5.5), esta seção apresenta um estudo de caso voltado à validação da proposta de melhoria de processo. A aplicação seguirá as seguintes etapas: inicialmente, será aplicado um questionário para avaliar o nível de conhecimento da equipe em relação às metodologias ágeis; em seguida, será realizada uma reunião explicativa sobre o funcionamento do novo processo. Após essa etapa, será iniciada a Sprint, e, ao final, será coletado o feedback da equipe. A Figura 10 apresenta o fluxo das atividades seguidas para a aplicação da proposta.

Figura 10 – Fluxo de aplicação da abordagem otimizada



Fonte: pelo autor.

### 5.6.1 Verificação do conhecimento do time

Antes de iniciar a aplicação da metodologia otimizada, foi realizada uma verificação do conhecimento da equipe em relação às metodologias ágeis, Scrum e Kanban. Esse diagnóstico teve como objetivo garantir que todos os membros envolvidos compreendessem os fundamentos da abordagem proposta e estivessem preparados para utilizar as práticas definidas na melhoria do processo.

A validação foi realizada por meio da aplicação de um questionário diagnóstico que se encontra no Apêndice C, contendo 6 questões objetivas, disponibilizado em formato de



formulário online via *google forms* no período de 26 de maio de 2025 a 30 de maio de 2025. Os dados coletados a partir dessa aplicação permitiram identificar o nível de conhecimento da equipe sobre as metodologias abordadas, servindo como base para os próximos passos da aplicação. Na Figura 11 segue as perguntas realizadas no questionário.

Figura 11 – Perguntas para validação de conhecimento do time.

Você conhece alguma metodologia ágil?
Quais dessas metodologias ou frameworks ágeis você conhece ou já teve contato (estudo ou prática)?
Já participou de algum projeto utilizando alguma metodologia ágil?
Qual é seu nível de familiaridade com o uso de quadros Kanban (como Trello, Jira, etc.)?
Você se sente confortável para trabalhar com metodologias ágeis?

Fonte: pelo autor.

### 5.6.2 Reunião explicativa do processo proposto

Após a aplicação do questionário diagnóstico, foi realizada uma reunião com os membros da equipe para apresentar a metodologia otimizada com base no Scrumban. O objetivo da reunião foi alinhar o time quanto ao funcionamento do novo processo, esclarecer dúvidas e garantir o entendimento sobre as práticas adotadas.

Durante a reunião, foram explicados os seguintes pontos:

- Visão geral da metodologia otimizada proposta;
- Etapas do fluxo de desenvolvimento mapeadas no diagrama de processo;
- Definição dos papéis dentro do time (analista de requisitos, líder técnico, desenvolvedor e analista de qualidade);
- Apresentação da ferramenta Trello e da estrutura do quadro Kanban a ser utilizado;
- Orientações sobre o monitoramento do cycle time, limitação de WIP e registro de defeitos.

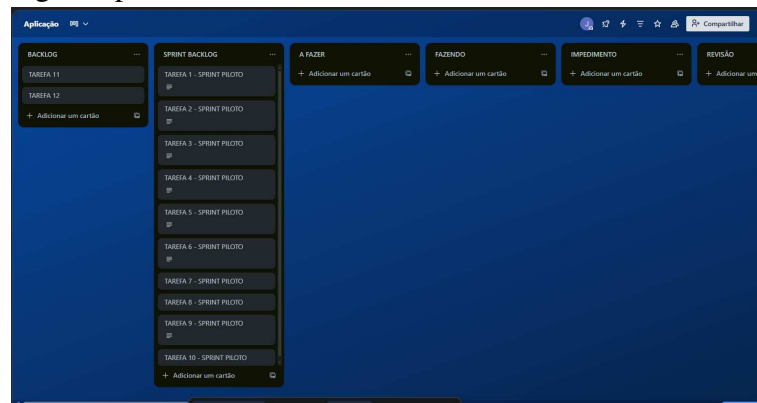
Essa reunião foi essencial para garantir o alinhamento do time e permitir o início da Sprint piloto de forma estruturada e conforme o processo proposto.

### 5.6.3 Aplicação do processo proposto

Com a equipe alinhada e o novo processo definido, foi iniciada a execução da *sprint* piloto utilizando a metodologia otimizada baseada no Scrumban. A *sprint* teve duração de uma semana e seguiu todas as etapas previstas no plano de aplicação.

Inicialmente, foi realizada uma reunião entre o analista de requisitos e o líder técnico para a definição dos itens do *Backlog* da *sprint*. Em seguida iniciou-se a etapa de refinamento, conduzida pelo analista de requisitos, que detalhou cuidadosamente os requisitos para evitar problemas de ambiguidade e dúvidas durante o desenvolvimento. A Figura 12 apresenta a execução da *sprint* com tarefas fictícias simuladas no Trello.

Figura 12 – Backlog da Sprint definido



Fonte: pelo autor.

Após a etapa de refinamento dos requisitos, a *sprint* teve início com a reunião de planejamento, na qual foram selecionadas as funcionalidades priorizadas a partir do *backlog* refinado. Cada demanda foi discutida com o time, e suas responsabilidades foram distribuídas entre os membros conforme os papéis definidos anteriormente. O time foi composto pelo líder técnico, analista de requisitos, dois desenvolvedores e analista de qualidade com supervisão geral do gerente de projetos.

Durante o ciclo de desenvolvimento, o time utilizou o quadro Kanban no Trello para organizar e acompanhar o progresso das tarefas. As colunas configuradas foram: *Backlog*, *Sprint Backlog*, *A Fazer*, *Em Andamento*, *Impedimentos*, *Revisão*, *Testes* e *Concluído*. Também foi inserida uma área para o registro de defeitos, com descrição do problema, causa e criticidade.

A equipe realizou reuniões diárias (*daily meetings*) com duração média de 10 a 15 minutos, nas quais foram discutidos o andamento das tarefas, bloqueios e o que seria feito no dia. Essa prática contribuiu para manter a equipe alinhada e identificar rapidamente possíveis

impedimentos.

Além disso, foi aplicado o conceito de inspeção contínua, no qual o líder técnico deve revisar os itens concluídos antes de passarem para a fase de testes. Esse processo permitiu corrigir falhas precocemente e garantir que os padrões de implementação estivessem sendo seguidos.

O *cycle time* de cada tarefa foi monitorado com base na movimentação dos cartões no quadro, desde a entrada na coluna "Em Andamento" até a chegada em "Concluído". Essa métrica permite analisar a eficiência do time e identificar gargalos no fluxo.

Ao final da sprint, foi realizada a reunião de *review*, na qual as entregas foram apresentadas e validadas. Posteriormente, foi conduzida a retrospectiva, onde o time discutiu os pontos positivos da Sprint e os aspectos que poderiam ser melhorados nas próximas iterações.

O planejamento inicial do estudo previa a aplicação do modelo proposto em duas sprints consecutivas, de forma a permitir a observação de possíveis ajustes de processo entre os ciclos. No entanto, devido às limitações de tempo para a execução do estudo de caso e à disponibilidade da equipe, foi possível realizar apenas uma sprint completa, que serviu como base para a análise dos resultados obtidos.

#### **5.6.4 *Feedback para validação do processo***

Com o objetivo de avaliar o desempenho do processo e a satisfação do time de desenvolvimento, foi aplicado um questionário contendo perguntas objetivas com escala de resposta entre 1 e 5, sendo 1 "discordo totalmente" e 5 "concordo totalmente". As perguntas foram divididas em seções, abordando os seguintes tópicos: planejamento e requisitos, execução e fluxo de trabalho, métricas e visibilidade, resultados percebidos, avaliação geral do processo. Além disso, incluiu-se uma seção com perguntas abertas para opiniões e sugestões de melhorias. O questionário utilizado encontra-se no Apêndice D.

## **6 RESULTADOS**

Neste capítulo é apresentado todos os resultados obtidos após a realização do estudo de caso no time. Portanto, esse capítulo está organizado da seguinte forma: Resultados do questionário para validação do conhecimento do time, resultados da Sprint pós aplicação do processo, resultados do feedback e validação dos objetivos específicos.

### **6.1 Resultados da entrevista com a EJ para compreensão do contexto da empresa**

A organização utilizada para a realização deste trabalho se trata de uma empresa júnior vinculada a Universidade Federal do Ceará, no campus de Russas, e é voltada para o desenvolvimento de soluções tecnológicas como sites, sistemas entre outros. A EJ atualmente conta com a participação de 15 pessoas na qual são discentes dos cursos de graduação em engenharia de software e ciência da computação. Além disso, a EJ é dividida em setores que são: presidência, vice-presidência, diretorias e time de desenvolvimento. Ao iniciar o desenvolvimento de um novo projeto é definida uma equipe composta por 5 a 6 membros, dependendo do tamanho do projeto, na qual dois são designers responsáveis pelo protótipo do sistema, dois são desenvolvedores e testadores e um responsável por gerenciar a equipe e alinhar os demais membros.

Atualmente a EJ possui 2 projetos em fase de desenvolvimento e diante disso foram levantados alguns questionamentos referentes aos problemas enfrentados pelo time de desenvolvimento. Diante disso foram relatados alguns problemas referentes a comunicação entre o time, auto-gerenciamento e o gerenciamento por parte do gerente de projetos.

No processo de desenvolvimento da EJ, inicialmente é realizada uma entrevista com o cliente para entender as necessidades do mesmo, possibilitando a elicitação de requisitos de software e criando um documento inicial de requisitos de acordo com as informações levantadas durante a etapa de entrevista. Em seguida o documento de requisitos é encaminhado para a etapa de prototipação na qual será compreendido os requisitos levantados e desenvolvido um protótipo inicial baseado nos requisitos.

Após finalizar a etapa de prototipação é realizada uma reunião com o cliente para que o protótipo elaborado seja validado, e assim garantir que atenda a real necessidade do cliente, nesta etapa os requisitos podem sofrer alterações por parte do cliente e assim um novo documento de requisitos pode ser gerado contemplando as alterações para que em seguida seja alterado o

protótipo, que será validado novamente pelo cliente. Após o protótipo ser aprovado é criado o *backlog* do produto com as demandas separadas e priorizadas para serem distribuídas para o time de desenvolvimento que inicia a codificação da demanda no começo da *Sprint*, que tem tempo de duração de 1 semana.

Após o final de cada sprint é realizado uma reunião com o cliente para que o mesmo realize a validação do que foi desenvolvido, caso não seja aprovado e o cliente dê sugestões de melhorias a demanda segue para a etapa de manutenção para ser corrigido e em seguida a demanda segue novamente para a etapa de validação. Caso a demanda seja aprovada ela é finalizada.

## **6.2 Resultados da entrevista com time de desenvolvimento para mapeamento de problemas**

Após uma análise das informações obtidas a partir da entrevista, foram identificados e mapeados problemas que impactam diretamente o desenvolvimento do software, ocasionando o atraso do projeto. A seguir estão dispostos os problemas encontrados no processo e como esses problemas influem para o atraso do projeto.

### *6.2.0.1 Processo de desenvolvimento de software não é bem definido*

O processo de desenvolvimento da EJ não é bem definido para a compreensão dos membros. Segundo informações coletadas, não existem documentações referente ao processo para que todos os membros permaneçam alinhados quanto aos objetivos e valores da organização.

Um processo de desenvolvimento que não é bem definido pode gerar consequências como o aumento dos custos para produção e manutenção do software e contribui para uma baixa qualidade do produto, o que pode levar ao atraso da entrega do projeto para o cliente. Além disso, um processo que não é bem definido pode levar a desmotivação da equipe de desenvolvimento, por falta de comunicação e dificuldade para gerenciar o time (XAVIER *et al.*, 2016).

### *6.2.0.2 Cerimônias importantes não são realizadas*

Visando acelerar a entrega de demandas e projetos de software, algumas cerimônias principais do Scrum, como as reuniões diárias e a revisão da sprint, foram reduzidas em frequência ou omitidas. Porém, a ausência dessas cerimônias pode acarretar diversas consequências

negativas para o projeto. Por exemplo, a falta de reuniões diárias pode resultar em desalinhamento da equipe de desenvolvimento em relação ao progresso das demandas, o que, por sua vez, dificulta a comunicação entre os membros da equipe e compromete o gerenciamento do projeto. Além disso, a falta da revisão da sprint pode impactar na qualidade do produto final, pois impede a avaliação do trabalho realizado durante a *sprint* e a identificação de possíveis melhorias. Como resultado, essa abordagem pode levar a atrasos na entrega e à diminuição da curva de aprendizado de toda a equipe de desenvolvimento (RASHEED *et al.*, 2021).

#### 6.2.0.3 *Falha na comunicação entre os membros do time*

A falta de comunicação eficaz entre os membros da equipe de desenvolvimento pode resultar em atrasos na entrega das demandas devido a conflitos entre o que foi desenvolvido e o que foi solicitado, levando à insatisfação do cliente. Além disso, uma comunicação deficiente dificulta o gerenciamento do time de desenvolvimento, pois não proporciona uma visão clara das atividades de cada membro durante a *Sprint*. Isso pode resultar em dificuldades para identificar e resolver problemas rapidamente, comprometendo ainda mais a eficiência e a qualidade do trabalho realizado. Considerando esses pontos, fica evidente que a deficiência na comunicação entre os membros da equipe está diretamente ligada à omissão das cerimônias do *Scrum*.

#### 6.2.0.4 *Não é realizado um gerenciamento de cronograma para o projeto*

Durante o desenvolvimento do projeto não é realizado um gerenciamento do cronograma estipulando prazos para entrega de demandas considerando sua complexidade, prioridade e riscos que podem surgir durante o desenvolvimento. A não realização deste cronograma pode implicar em alguns problemas como a dificuldade para gerenciar o time e controlar o desenvolvimento do projeto, o que pode levar a atrasos na entrega, por não considerarem vários fatores durante o desenvolvimento (LIMA, 2022).

#### 6.2.0.5 *Não são realizadas verificações e validações durante o desenvolvimento do projeto*

Durante a etapa de desenvolvimento do software não são realizadas revisões e testes do código-fonte, o que pode levar a uma perda da qualidade do produto, visto que a probabilidade de se encontrar falhas e bugs no produto são bem maiores, acarretando atrasos na entrega de demandas e do projeto em si. Após estudos realizados pela **SOFTWARE EXcellence** (SOFTEX),

cerca de 40% a 50% dos esforços gastos no desenvolvimento de projetos se dá devido ao retrabalho necessário para realizar correções de bugs e falhas no sistema(SOFTEX, 2011).

#### *6.2.0.6 Problemas com mudanças nos requisitos do projeto*

Durante o desenvolvimento não é realizado um planejamento de alterações no escopo da demanda ou do projeto. Além disso, conforme analisado no processo, percebe-se que o cliente não tem uma participação ativa durante o desenvolvimento, geralmente o mesmo participa apenas durante a validação do protótipo e durante a validação final. O que pode levar ao atraso da entrega do projeto, visto que o cliente possa solicitar mudanças (ESPINDOLA *et al.*, 2004).

#### *6.2.0.7 Não são registrados a quantidade de bugs e falhas encontradas*

Durante o desenvolvimento do projeto não são registrados a quantidade de bugs e falhas encontradas, que são importantes para identificar pontos de melhoria contínua e utilizar como métrica para medir a qualidade do processo e do produto.

#### *6.2.0.8 Não é realizado tratamento de requisitos inconsistentes ou ambíguos*

Os requisitos documentados após o processo de elicitação de requisitos com o cliente não são tratados de modo a evitar inconsistências ou ambiguidade ocasionando conflitos e divergências entre o que foi solicitado pelo cliente e o que foi desenvolvido. Não realizar o tratamento dos requisitos de forma adequada pode levar a não conformidade do sistema, gerando retrabalho para realizar as correções e levando ao atraso dos projetos (LIMA, 2022).

#### *6.2.0.9 Dificuldade de gerenciamento da equipe*

Quando são atribuídas atividades para os membros do time de desenvolvimento, o diretor de projetos possui dificuldade para gerenciar os itens que estão sendo desenvolvidos, bem como o progresso de cada colaborador.

### **6.3 Resultados do mapeamento das práticas do Scrumban**

Como mencionado na fundamentação teórica desta pesquisa, a metodologia *Scrumban* é flexível e não estabelece um conjunto de práticas de maneira fixa. Portanto, foram mapeados a

seguintes práticas que estavam alinhadas com os problemas enfrentados pela EJ.

#### *6.3.0.1 Definição de backlog e refinamento*

Criar o *backlog* do produto e realizar a priorização das tarefas contribui para que todo o time se mantenha alinhado com os objetivos do projeto em andamento. Essa gestão do *backlog* permite que as equipes se adaptem de forma mais eficaz em relação as mudanças nos requisitos, assegurando que as entregas mais importantes sejam realizadas primeiro (FERREIRA *et al.*, 2023).

#### *6.3.0.2 Reuniões diárias*

Estabelece uma melhor comunicação rápida entre o time para identificar possíveis impedimentos e alinhar sobre o progresso no decorrer da Sprint. Estas reuniões promovem transparência e colaboração entre os membros, o que permite a captação de problemas futuros e ajuda na gestão do time (DINIZ, 2017).

#### *6.3.0.3 Revisão da Sprint*

Segundo o Guia do Scrum<sup>1</sup> a revisão da Sprint promove a cultura do feedback contínuo e valida as entregas realizadas, garantindo que o produto atenda às expectativas dos *Stakeholders* e possibilite ajustes de acordo com a necessidade. Esta prática contribui para resolução de problemas relacionados a mudanças nos requisitos de forma dinâmica e eficaz, permitindo que os membros da equipe se adapte rapidamente às novas demandas sem implicar na qualidade do produto.

#### *6.3.0.4 Retrospectiva da Sprint*

Promove a melhoria continua baseada nas experiências vivenciadas no decorrer da Sprint. Contribuí para a identificação de pontos que funcionaram bem e os que precisam melhorar e buscando tratativas para aplicar na próxima sprint (DINIZ, 2017).

---

<sup>1</sup> <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf>



#### 6.3.0.5 *Planejamento da Sprint*

Garante um melhor alinhamento do time em relação as demandas que serão realizadas e que todos os pontos sejam repassados corretamente. Além disso, contribui que as expectativas e metas para a Sprint sejam claras e realistas (SASSA *et al.*, 2023).

#### 6.3.0.6 *Visualização do fluxo de trabalho*

A utilização do quadro kanban é uma prática que permite uma maior transparência e visualização do fluxo de trabalho para toda a equipe, o que contribui para a identificação de gargalos no processo e garante a compreensão do estado atual do trabalho (LOPES *et al.*, 2024).

#### 6.3.0.7 *Revisão contínua do fluxo de trabalho*

Realizar um monitoramento constante do fluxo de trabalho contribui para a identificação de gargalos na esteira que podem ser otimizados, diminuindo o tempo de ciclo da demanda e promovendo um aumento da produtividade (FERREIRA *et al.*, 2023).

#### 6.3.0.8 *Monitoramento do tempo de ciclo (Cycle Time)*

Monitorar o tempo que as tarefas levam até chegar ao final do ciclo é muito importante, pois permite medir a eficiência do processo e a produtividade da equipe de desenvolvimento. Esta métrica pode ser utilizada para a identificação de gargalos no processo e instigar a busca por soluções que possam mitigar e melhorar o processo.

#### 6.3.0.9 *Priorização dinâmica no quadro kanban*

Esta prática permite que as demandas sejam reorganizadas para atender as mudanças do projeto, garantindo que as demandas mais críticas sejam realizadas primeiro e contribui para uma melhor adaptação do time de desenvolvimento em relação as novas prioridades no ambiente ágil melhorando a eficiência e entrega de valor ao cliente (ANDERSON; CARMICHAEL, 2016).

#### 6.3.0.10 *Métricas de defeitos no kanban*

As métricas de defeitos ajudam a monitorar a qualidade tanto do processo, quanto do produto em desenvolvimento. Estas métricas podem indicar a frequência e recorrência de falhas

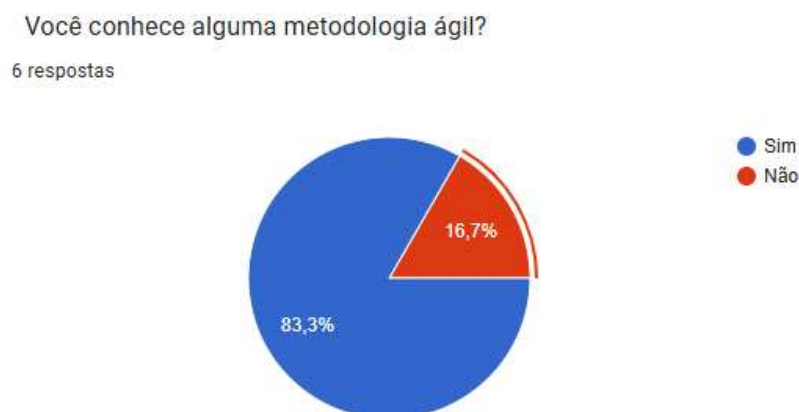
que ocorreram ao longo do fluxo de desenvolvimento. Além disso, estas métricas podem ser utilizadas como parâmetros para analisar mudanças no processo ou no time de desenvolvimento.

#### 6.4 Resultados do questionário para validação de conhecimento do time

Após o período estabelecido para o preenchimento do questionário diagnóstico, foram obtidas seis respostas, provenientes do gerente de projetos, líder técnico, analista de requisitos e membros do time de desenvolvimento. Além disso, as respostas foram organizadas por categorias, refletindo os principais aspectos abordados na metodologia otimizada. Com a análise dos dados foi possível verificar a aderência das práticas adotadas, identificar pontos positivos e possíveis melhorias a serem consideradas, além de contribuir com a validação dos objetivos propostos nesse trabalho

A análise das respostas revelou que entre os seis participantes, apenas um declarou não possuir conhecimento prévio sobre metodologias ágeis. Esse resultado evidencia uma equipe com níveis de conhecimento heterogêneos, indicando a necessidade de realizar um nivelamento conceitual com o membro em questão, a fim de garantir uma compreensão mínima comum sobre os fundamentos da abordagem proposta. A figura 13 apresenta o gráfico para melhor visualização das informações abordadas.

Figura 13 – Resposta da questão 1 do questionário



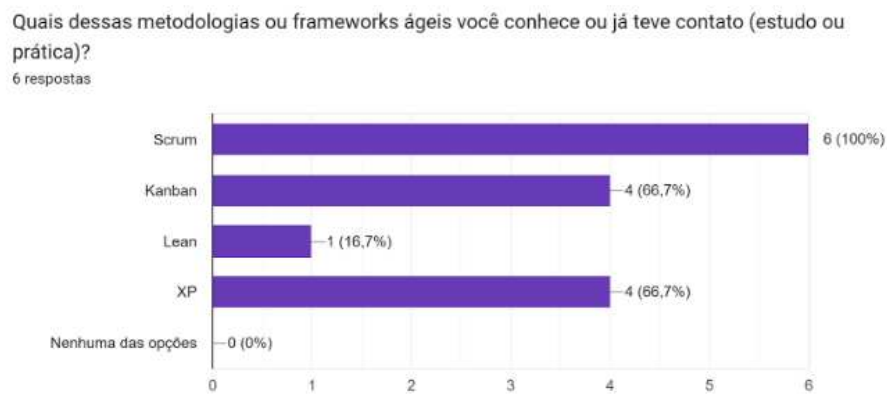
Fonte: pelo autor.

Contudo, na segunda questão, ao serem apresentadas opções de metodologias ágeis, todos os participantes indicaram já conhecer ou ter tido contato com ao menos uma delas. Esse resultado reforça que a equipe possui uma base estruturada de conhecimento nesse tipo de abordagem, o que favorece a adoção da abordagem proposta neste trabalho.

Além disso, na terceira questão, foi constatado que mais da metade dos membros já participou ativamente de projetos utilizando metodologias ágeis, o que fortalece ainda mais a viabilidade de aplicação da proposta.

As Figuras 14 e 15 apresentam, respectivamente, os gráficos referentes às metodologias que os membros da equipe conhecem ou já tiveram contato, e à experiência prática com projetos ágeis.

Figura 14 – Resposta da questão 2 do questionário



Fonte: pelo autor.

Figura 15 – Resposta da questão 3 do questionário



Fonte: pelo autor.

Contudo, após a análise dos resultados da pesquisa, foi possível concluir que todo o time detem conhecimentos sólidos em relação a metodologias ágeis, o que favorece a aplicação da proposta sem a necessidade de treinamento. Dessa forma, torna-se viável dar continuidade à aplicação prática da metodologia otimizada, conforme o plano estabelecido.

## **6.5 Resultados da Sprint pós aplicação do processo**

Durante a sprint piloto, realizada no período de 14/07/2025 a 18/07/2025, durante a reunião de planejamento foram priorizadas 10 atividades para serem desenvolvidas pelo time de desenvolvimento da empresa júnior. Ao final da iteração, 8 atividades foram concluídas com sucesso e as duas atividades restantes foram postergadas para a Sprint posterior devido a ocorrência de bugs.

O tempo de ciclo médio registrado foi de aproximadamente 1 dia e meio por item, considerando que as demandas abordadas foram de baixa complexidade e exigiram testes mais simples e rápidos. Portanto, pode-se destacar que o processo teve um impacto positivo em itens menos complexos, o que permitiu entregas mais rápidas e contínuas.

Além disso, durante a execução da Sprint, dois defeitos foram identificados no decorrer da etapa de testes, o que impactou na entrega de duas atividades. Tais defeitos encontrados evidenciam a importância das etapas de revisão das atividades e da etapa de validação.

O quadro Kanban utilizado contribuiu para uma melhor visibilidade das atividades e facilitou o acompanhamento do fluxo de trabalho. Além disso, as reuniões diárias foram realizadas corretamente, auxiliando o alinhamento das atividades e na identificação e resolução de impedimentos.

A EJ não realiza a marcação de métricas como *bugs* ou *cycle time* para avaliação do processo utilizado. Portanto, não foi possível realizar um comparativo com Sprints anteriores devido a inexistência desses dados. A figura 16 apresenta o gráfico das tarefas que foram planejadas para a Sprint e as que foram concluídas dentro do prazo.

Figura 16 – Tarefas planejadas x Tarefas concluídas



Fonte: pelo autor.

Apesar da Sprint conter apenas itens de baixa complexidade, os resultados que foram obtidos demonstraram que a aplicação do processo proposto possibilitou maior controle, organização e agilidade no desenvolvimento. Os dados obtidos com a aplicação servem como parâmetros para comparações em Sprints futuras com itens mais complexos, podendo avaliar a eficácia da abordagem otimizada.

## 6.6 Resultados do questionário de feedback pós aplicação do processo

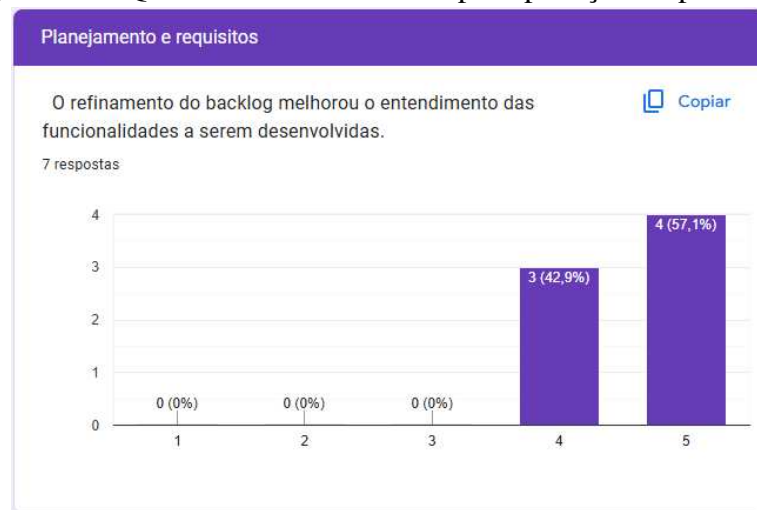
O questionário de feedback pós-aplicação do novo processo foi respondido pelos membros da equipe de desenvolvimento da empresa júnior, incluindo desenvolvedores e gestores. As perguntas abordaram temas como satisfação geral, facilidade de adaptação às mudanças e sugestões para aprimorar o trabalho em equipe. Partindo da hipótese de que as melhorias implementadas no processo aumentariam a satisfação e o engajamento dos participantes, a análise das respostas confirmou essa expectativa: houve alta aceitação das mudanças, com destaque para melhorias na organização, comunicação e clareza das tarefas, além de sugestões construtivas para evoluir ainda mais o processo.

### 6.6.1 Planejamento e requisitos

Todas as respostas ficaram entre 4 e 5, indicando que tanto o refinamento do backlog e a priorização dinâmica ajudaram no entendimento das funcionalidades e na organização

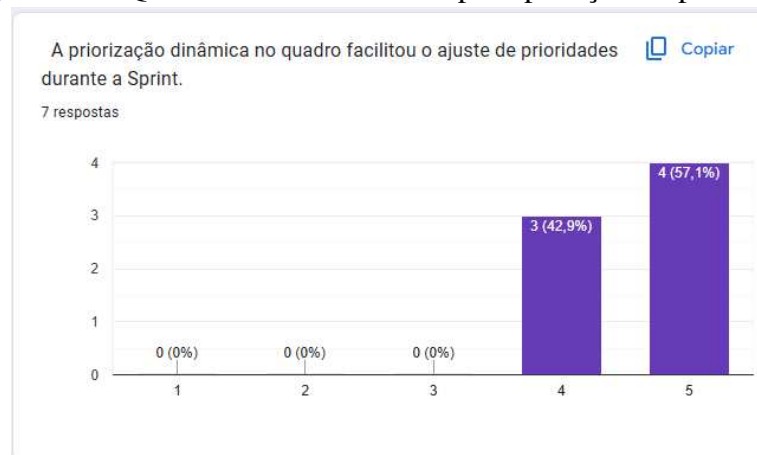
das entregas. Com um melhor detalhamento das funcionalidades foi possível convergir o entendimento das atividades priorizadas durante a Sprint, evitando ambiguidades e reduzindo custos com retrabalho. Além disso, com a visualização do quadro Kanban houve uma melhor organização e fácil compreensão quanto as priorizações dinâmicas no decorrer da Sprint. As figuras 17 e 18 abordam as respostas dos integrantes sobre essas práticas e como elas foram bem recebidas pelo time.

Figura 17 – Pergunta 01 - Questionário de feedback pós aplicação do processo



Fonte: pelo autor.

Figura 18 – Pergunta 02 - Questionário de feedback pós aplicação do processo

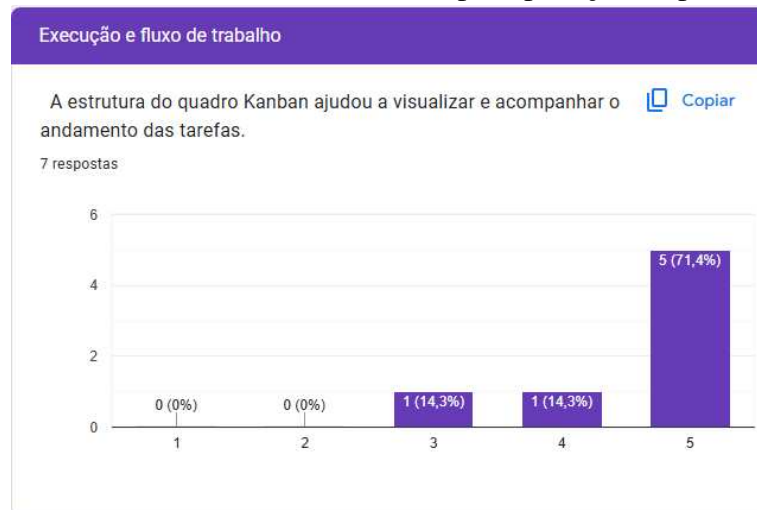


Fonte: pelo autor.

### 6.6.2 Execução e fluxo de trabalho

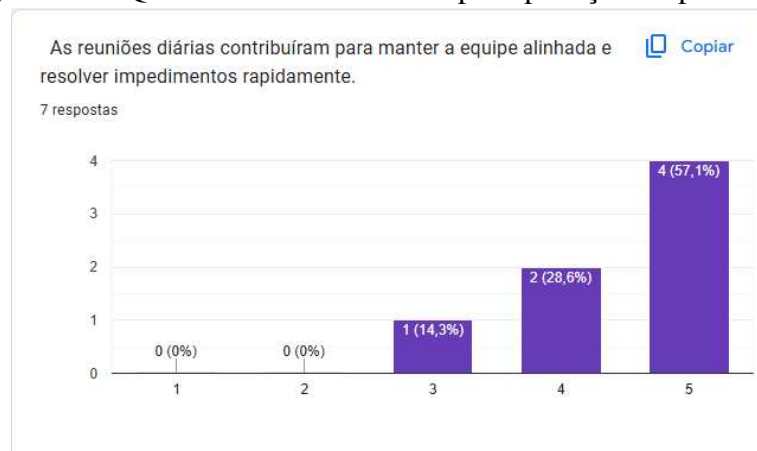
A estrutura do quadro Kanban e as reuniões diárias foram bem avaliadas, com maioria das respostas entre 4 e 5, ou seja, a visualização do quadro e as reuniões diárias tornaram o processo mais claro e fácil de compreender e acompanhar as atividades em progresso, contribuindo para maior produtividade e proatividade do time de desenvolvimento. No entanto, houveram respostas medianas, o que indica aberturas para melhorias quanto ao quadro Kanban e ao processo. As figuras 19 e 20 abordam a satisfação dos membros em relação quadro Kanban e as reuniões diárias realizadas durante a Sprint piloto.

Figura 19 – Pergunta 04 - Questionário de feedback pós aplicação do processo



Fonte: pelo autor.

Figura 20 – Pergunta 05 - Questionário de feedback pós aplicação do processo

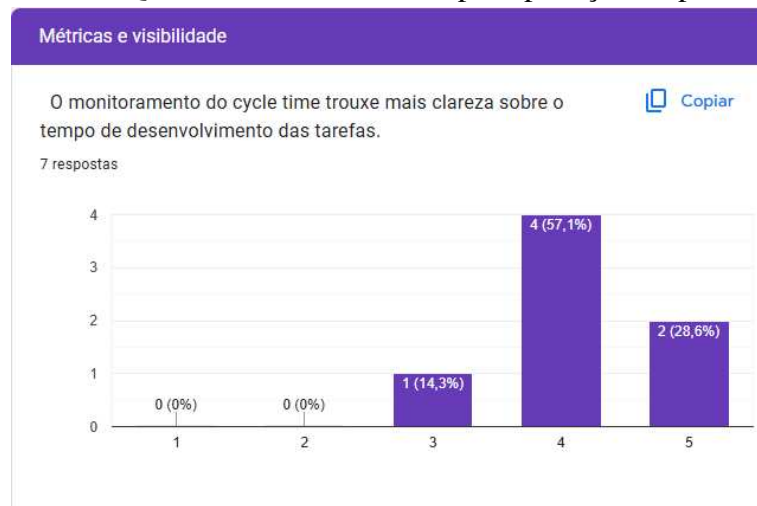


Fonte: pelo autor.

### 6.6.3 Métricas e visibilidade

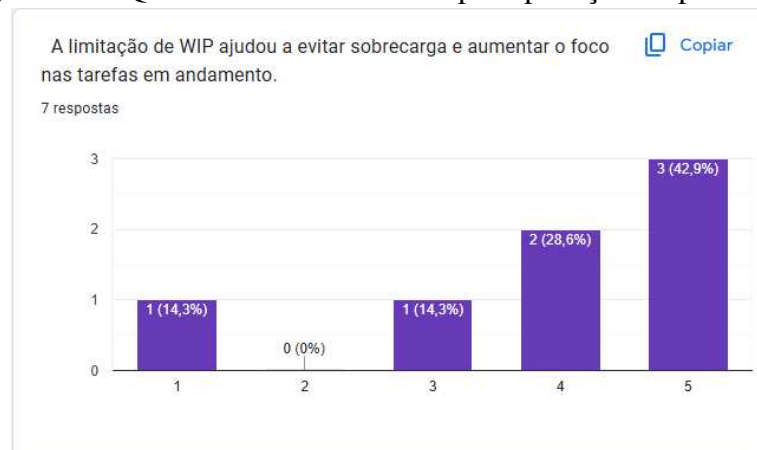
O monitoramento do *cycle time* e a limitação do WIP foram aceitas de forma considerável pelo time, possivelmente devido ao baixo impacto no cotidiano da equipe. Com a análise das respostas foi possível perceber que a maioria das notas estão entre 4 e 5 indicando que o time compreendeu a importância de manter o registro de métricas para otimização de Sprints futuras e evitar sobrecarga de trabalho da equipe. No entanto houveram notas insatisfeitas e intermediárias, o que abre discussões para trabalhar em melhorias quanto a essas métricas. As figuras 21 e 22 demonstram as respostas dos membros em relação as métricas, houve algumas divergências nas respostas o que deixa aberto a discussões futuras.

Figura 21 – Pergunta 08 - Questionário de feedback pós aplicação do processo



Fonte: pelo autor.

Figura 22 – Pergunta 09 - Questionário de feedback pós aplicação do processo



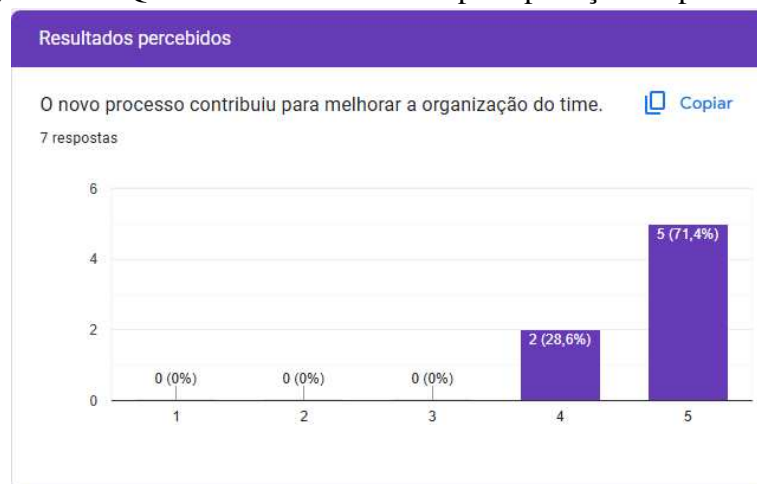
Fonte: pelo autor.



#### 6.6.4 Resultados percebidos

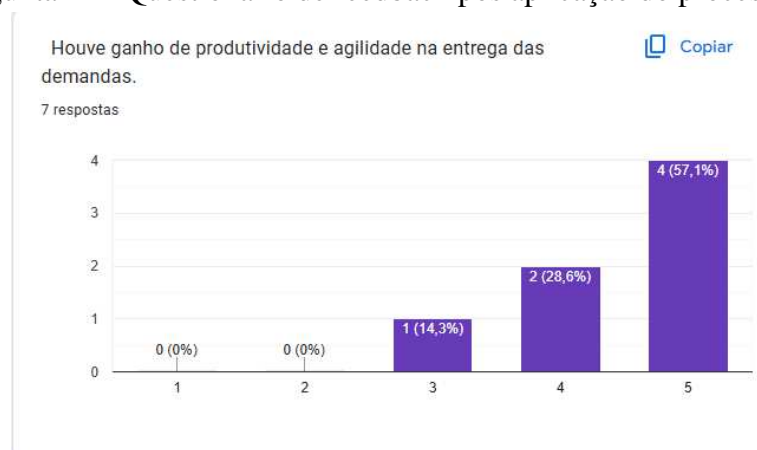
Alguns resultados percebidos pelo time foi que as melhorias efetuadas no processo, tiveram um efeito positivo contribuindo para uma melhor organização do time, redução de falhas e aumento de produtividade e comunicação. Analisando os gráficos nas Figuras 23 e 24 é possível perceber que a maior parte da equipe considera que com as melhorias implementadas tiveram impacto significativo, no entanto houve uma das respostas indicou nota mediana, ou seja, apenas um membro não considerou que as melhorias tiveram impacto positivo.

Figura 23 – Pergunta 11 - Questionário de feedback pós aplicação do processo



Fonte: pelo autor.

Figura 24 – Pergunta 12 - Questionário de feedback pós aplicação do processo

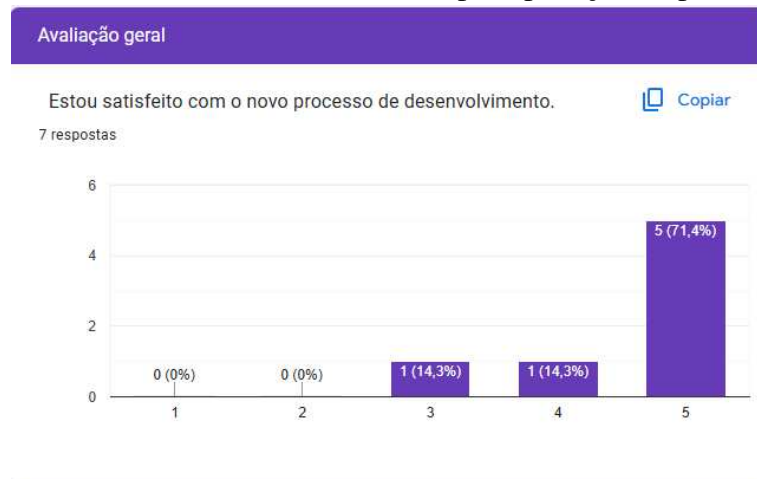


Fonte: pelo autor.

### 6.6.5 Avaliação geral

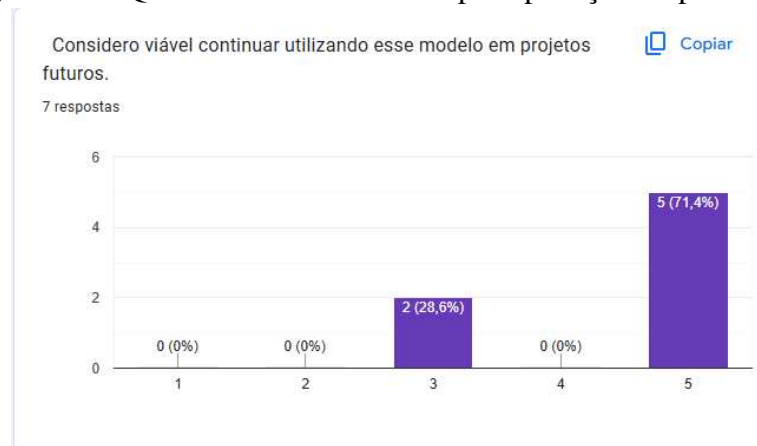
Contudo, analisando de forma geral a maioria dos membros da equipe consideram viável continuar utilizando esse processo em projetos futuros, portanto é possível inferir que com base nas respostas positivas e na disposição da equipe em adotar o processo em projetos futuros, é possível afirmar que a metodologia otimizada apresentou resultados satisfatórios e viabilidade de utilização em projetos futuros. Contudo, houve algumas notas interpostas, indicando a abertura para continuar evoluindo o processo. As Figuras 25 e 26 apresentam as respostas dos usuários quanto satisfação com o processo melhorado e viabilidade de utilização em projetos futuros.

Figura 25 – Pergunta 14 - Questionário de feedback pós aplicação do processo



Fonte: pelo autor.

Figura 26 – Pergunta 15 - Questionário de feedback pós aplicação do processo

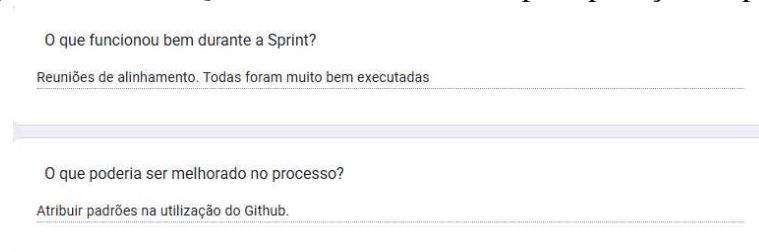


Fonte: pelo autor.

### 6.6.6 Questionamentos abertos e sugestões

Para os questionamentos abertos, alguns participantes deram feedbacks positivos sobre o que funcionou bem durante a Sprint piloto. Além disso, um dos participantes sugeriu a padronização de uso do GitHub pela equipe. Este pode ser considerado um *insight* valioso, pois possibilita a melhoria contínua, especialmente no que diz respeito à organização e rastreabilidade do desenvolvimento. A Figura 26 apresenta a resposta dos questionamentos abertos.

Figura 27 – Pergunta 17 e 18 - Questionário de feedback pós aplicação do processo



O que funcionou bem durante a Sprint?

Reuniões de alinhamento. Todas foram muito bem executadas

---

O que poderia ser melhorado no processo?

Atribuir padrões na utilização do Github.

Fonte: pelo autor.

## 6.7 Validação dos objetivos

Após a aplicação prática da metodologia otimizada e nos resultados obtidos ao longo do trabalho, é possível validar o atendimento aos objetivos específicos propostos inicialmente. A seguir, apresenta-se a análise de cada um deles:

- Analisar e entender o processo de desenvolvimento existente na empresa júnior: este objetivo foi atingido na seção 5.3, onde foi descrito e analisado o processo atual da equipe, permitindo a identificação problemas críticos e gargalos que afetavam a produtividade e a qualidade das entregas.
- Identificar problemas recorrentes no processo de desenvolvimento: Os principais problemas foram levantados a partir da observação prática e da análise do processo descrito, sendo discutidos nas seções 5.3.1 e 5.4, e utilizadas como base para o mapeamento das melhorias.
- Relacionar as dificuldades encontradas com as boas práticas da metodologia híbrida Scrumban: a relação foi estabelecida na seção 5.4 onde as práticas do Scrumban foram analisadas e escolhidas conforme a aderência aos problemas identificados no processo atual.
- Elaborar melhorias no processo baseado nas práticas do Scrumban: As melhorias propostas

estão descritas na seção 5.5, sendo detalhadas e justificadas conforme sua aderência aos problemas encontrados e seu potencial de impacto positivo na organização do trabalho.

- Validar o modelo proposto e coletar seu feedback: A validação foi realizada na aplicação da Sprint piloto (5.6) e por meio do questionário de feedback (seção .1). Os resultados demonstraram a aceitação da equipe, a efetividade das melhorias propostas e o potencial de adoção do processo otimizado em futuros projetos.

Contudo, pode-se concluir que todos os objetivos específicos estabelecidos foram atingidos com sucesso ao longo do desenvolvimento deste trabalho.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Em conclusão deste trabalho, ficou evidente a importância de se utilizar metodologias ágeis para gerenciar projetos, devido a sua flexibilidade e fácil adaptação seguindo a necessidade de cada projeto. No entanto, trabalhar com apenas uma metodologia ágil pode conter limitações que impactam a produtividade, organização e a comunicação na empresa (OLIVEIRA; PEDRON, 2021). Em detrimento a isso, este trabalho teve como objetivo propor melhorias no processo de desenvolvimento de software adotado por uma empresa júnior, a partir da aplicação de práticas baseadas na metodologia híbrida Scrumban. A proposta visou tornar o processo mais eficiente, ágil e visual, para promover uma melhor organização, comunicação e controle do fluxo de trabalho durante os ciclos de desenvolvimento.

Para a realização deste trabalho foi necessário realizar uma análise do processo atual da EJ, onde foram identificados diversos problemas como: baixa clareza dos requisitos, ausência de priorização dinâmica, retrabalhos frequentes, ausência de métricas e dificuldades para acompanhar o progresso das atividades. Com base nessas problemáticas, foi realizado um mapeamento de práticas da metodologia Scrumban, e posteriormente uma proposta de melhoria estruturada e que envolveu a definição de papéis, introdução do refinamento de backlog, organização do fluxo com quadro Kanban, adoção de métricas como cycle time e WIP, além da implementação de cerimônias como reuniões diárias, review e retrospectiva.

A metodologia otimizada foi aplicada em uma Sprint piloto, que ocorreu após a realização de um diagnóstico para validar o conhecimento do time e uma reunião de alinhamento sobre o novo processo. Os resultados da Sprint demonstraram melhorias significativas na organização das atividades, maior clareza nas entregas e melhor comunicação entre os membros. O feedback do time, coletado por meio de questionário, reforçou a boa aceitação das mudanças, destacando como pontos fortes o refinamento do backlog, a visualização do quadro Kanban e o uso de métricas, ainda que com sugestões para melhorias futuras.

Ademais, o trabalho contribuiu com um modelo prático de melhoria de processo que pode ser adotado ou adaptado por outras equipes de desenvolvimento com estrutura semelhante. A experiência mostrou que embora o ambiente possua recursos limitados, a adoção de práticas simples e bem definidas pode gerar impactos positivos na produtividade e qualidade do produto final. Além disso, este modelo pode ser adaptado para outras EJs com um contexto semelhante.

Como limitação, pode-se destacar que a aplicação foi feita em apenas uma Sprint e com tarefas de baixa complexidade, o que restringe a avaliação em contextos de maior escala.

Portanto, para trabalhos futuros, recomenda-se a continuidade da aplicação em múltiplas Sprints, avaliação das métricas ao longo do tempo, e ajustes nas práticas conforme a maturidade e feedback da equipe. Além disso, buscar práticas em outras metodologias que possam agregar ao processo e otimizar ainda mais a produtividade e organização.

Dessa forma, pode-se concluir que o modelo proposto atendeu aos objetivos traçados, validando a viabilidade da aplicação da metodologia Scrumban como alternativa para melhorar o processo de desenvolvimento de software em empresas júniores.

## REFERÊNCIAS

- ABRAHAMSSON, P. e. a. New directions on agile methods: a comparative analysis. **25th International Conference on Software Engineering, 2003**, 2003.
- AGRAWAL, M.; CHARI, K. Software effort, quality, and cycle time: A study of cmm level 5 projects. **IEEE Transactions on software engineering**, IEEE, v. 33, n. 3, p. 145–156, 2007.
- AHMAD, M. O.; MARKKULA, J.; OIVO, M. Kanban in software development: A systematic literature review. In: IEEE. **2013 39th Euromicro conference on software engineering and advanced applications**. [S. l.], 2013. p. 9–16.
- ALQUDAH, M.; RAZALI, R. An empirical study of scrumban formation based on the selection of scrum and kanban practices. **Int. J. Adv. Sci. Eng. Inf. Technol**, v. 8, n. 6, p. 2315–2322, 2018.
- ALVES, D. d. S. Estudo de caso: o uso de user stories aplicado na construção de um backlog de produto. 2022.
- ANDERSON, D. J. **Kanban: successful evolutionary change for your technology business**. [S. l.]: Blue hole press, 2010.
- ANDERSON, D. J.; CARMICHAEL, A. **Kanban esencial condensado**. [S. l.]: PocketBook4you, 2016.
- BHAVSAR, K.; SHAN, V.; GOPALAN, S. Scrumbanfall: an agile integration of scrum and kanban with waterfall in software engineering. **International Journal of Innovative Technology and Exploring Engineering (IJITEE)**, 2020.
- BISSI, W. Metodologia de desenvolvimento ágil. **Campo Digital**, 2007.
- DIAS, G. A. **Estudo de caso: aplicação de uma proposta de combinação metodologias ágeis com ênfase em gestão da inovação**. Dissertação (Mestrado) – Universidade Federal do Ceará, Campus de Russas, 2022.
- DINIZ, D. F. M. Um estudo sobre as principais dificuldades de aplicação do scrum no gerenciamento de projetos de software. 2017.
- ESPINDOLA, R. S. de; MAJDENBAUM, A.; AUDY, J. L. N. Uma análise crítica dos desafios para engenharia de requisitos em manutenção de software. In: **WER**. [S. l.: s. n.], 2004. p. 226–238.
- FERREIRA, M. S. *et al.* Ferramentas ágeis no desenvolvimento de projetos por engenheiros químicos: uma revisão sobre lean, scrum e kanban. Universidade Federal de Uberlândia, 2023.
- FOWLER, M.; HIGHSMITH, J. The agile manifesto. **Software development**, 2001.
- FUGGETTA, A. Software process: a roadmap. **Proceedings of the Conference on the Future of Software Engineering**, 2000.
- KHAN, Z. A. **Scrumban - Adaptive Agile Development Process Using scrumban to improve software development process**. 2014.

- LADAS, C. **Scrumban-essays on kanban systems for lean software development**. [S. l.]: Lulu.com, 2009.
- LIKER, J. **O modelo Toyota–Princípios de gestão**. [S. l.]: Porto Alegre: Bookman, 2004.
- LIKER, J. K.; HOSEUS, M. Human resource development in toyota culture. **International Journal of Human Resources Development and Management**, 2010.
- LIMA, E. B. d. **Gerenciamento de projetos de software: uma pesquisa mercadológica**. Dissertação (B.S. thesis) – Universidade Federal do Rio Grande do Norte, 2022.
- LOPES, L.; SOUZA, G.; ALVES, A.; NUNES, R. Metodologias Ágeis: Explorando o impacto do scrum e do kanban na qualidade e produtividade do software. **Revista Multidisciplinar do Nordeste Mineiro**, v. 12, 11 2024.
- MARTINHAGO, F. B. Utilização de modelos híbridos para a definição e execução de processos de software aplicados na otimização da produtividade. 2018.
- NÉSIO, R.; FREITAS, P. F.; FREITAS, G. M. Laboratório de projeto-concepção, gestao e implementação de projetos aplicados ao ensino de engenharia de sistemas. v. 2, n. 1, 2020.
- OLIVEIRA, R. L. F.; PEDRON, C. D. Métodos ágeis: Uma revisão sistemática sobre benefícios e limitações. **Brazilian Journal of Development**, v. 7, n. 1, p. 4520–4534, 2021.
- PATILLA, H. J.; ALVIAR, J. M. V.; CONISLLA, Y. M. Scrumban/xp: Propuesta para mejorar la eficiencia de la gestión de proyectos ágiles en el desarrollo de software. **Revista Ibérica de Sistemas e Tecnologias de Informação**, Associação Ibérica de Sistemas e Tecnologias de Informacao, n. E61, p. 14–32, 2023.
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software-9**. [S. l.]: McGraw Hill Brasil, 2021.
- RASHEED, A.; ZAFAR, B.; SHEHRYAR, T.; ASLAM, N. A.; SAJID, M.; ALI, N.; DAR, S. H.; KHALID, S. Requirement engineering challenges in agile software development. **Mathematical Problems in Engineering**, Hindawi Limited, v. 2021, p. 1–18, 2021.
- REDDY, A. The scrumban [r] evolution: getting the most out of agile, scrum, and lean kanban. **Addison-Wesley Professional**, 2015.
- REIS, A. A. Scrumban-metodologia híbrida com scrum e kanban para desenvolvimento de software. 002, 2021.
- RISING, L.; JANOFF, N. S. The scrum software development process for small teams. **IEEE software**, 2000.
- ROYCE, W. W. Managing the development of large software systems: concepts and techniques. **Proceedings of the 9th international conference on Software Engineering**, 1987.
- SASSA, A. C.; ALMEIDA, I. A. de; PEREIRA, T. N. F.; OLIVEIRA, M. S. de. Scrum: A systematic literature review. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 14, n. 4, 2023.
- SCHWABER, K.; SUTHERLAND, J. The scrum guide. **Scrum Alliance**, v. 21, n. 1, p. 1–38, 2011.



SOARES, M. D. S. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação**, 2004.

SOFTEX, M. Br-melhoria de processo do software brasileiro: Guia geral. **Brasília: Associação para Promoção da Excelência do Software Brasileiro**, 2011.

SOMMERVILLE, I. **Software engineering (ed.)**. [S. l.]: America: Pearson Education Inc, 2011.

VASCONCELOS, Y. L. Metodologia scrum e a gestão de atividades didático-pedagógicas realizadas em colaboração. **Revista Tecnologias na Educação–Ano**, 2016.

XAVIER, D.; VIANA, D.; GADELHA, B. Um estudo sobre a relação entre processo e motivação em equipes de desenvolvimento de software. In: SBC. **Anais do XII Simpósio Brasileiro de Sistemas de Informação**. [S. l.], 2016. p. 076–083.

## **APÊNDICE A - ROTEIRO DA ENTREVISTA COM A EJ**

- Como está organizada a equipe da empresa júnior atualmente?
- Quais funções ou cargos existem dentro da equipe e como são distribuídas as responsabilidades?
- Como é feita a comunicação entre os membros da equipe?
- Como é feito o planejamento e acompanhamento dos projetos na empresa júnior?
- Quais são os principais desafios que a equipe enfrenta durante o desenvolvimento dos projetos?

**APÊNDICE B - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO**

## TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

---

Título da Pesquisa: “Proposta de Otimização do Modelo de Processo de uma Empresa Júnior pela Utilização da Abordagem Híbrida Scrumban”

Pesquisador: Antonio Juliano Fernandes Rocha Pinheiro – Discente do curso de Graduação em Engenharia de Software

### **Informações ao Participante**

Você está sendo convidado(a) a participar de uma pesquisa acadêmica. Sua participação é voluntária e você pode desistir a qualquer momento, sem nenhum prejuízo.

O estudo busca entender práticas e propor melhorias nos processos de uma Empresa Júnior utilizando a abordagem híbrida Scrumban. Para isso, poderá ser solicitado que você responda questionários ou participe de entrevistas.

Todas as informações fornecidas serão confidenciais e utilizadas apenas para fins acadêmicos.

### **Declaração de Consentimento**

Declaro que recebi informações sobre a pesquisa, compreendi seus objetivos e concordo em participar voluntariamente.

Assinatura do(a) Participante: \_\_\_\_\_

Data: \_\_\_\_/\_\_\_\_/\_\_\_\_

Assinatura do Pesquisador: \_\_\_\_\_

## **APÊNDICE C - QUESTIONÁRIO PARA VALIDAÇÃO DE CONHECIMENTO**

## Questionário – Validação de Conhecimento em Metodologias Ágeis

---

### 1. Informe seu curso \*

- ☐ Engenharia de Software
  - ☐ Ciência da Computação
- 

### 2. Qual semestre está cursando? \*

- ☐ 1º ☐ 2º ☐ 3º ☐ 4º ☐ 5º ☐ 6º ☐ 7º ☐ 8º
- 

### 3. Você conhece alguma metodologia ágil? \*

- ☐ Sim
  - ☐ Não
- 

### 4. Quais dessas metodologias ou frameworks ágeis você conhece ou já teve contato (estudo ou prática)? \*

- ☒ Marque todas que se aplicam:
  - ☐ Scrum
  - ☐ Kanban
  - ☐ Lean
  - ☐ XP
  - ☐ Nenhuma das opções
- 

### 5. Já participou de algum projeto utilizando alguma metodologia ágil? \*

- ☐ Sim, participei ativamente
  - ☐ Sim, mas com pouca participação
  - ☐ Não
- 

### 6. Qual sua função na equipe? \*

- ☐ Desenvolvedor
  - ☐ Líder Técnico
  - ☐ Analista de Requisitos
  - ☐ Analista de Qualidade
- 

### 7. Qual é seu nível de familiaridade com o uso de quadros Kanban (como Trello, Jira, etc.)? \*

- ☐ Já utilizei em projetos
  - ☐ Conheço, mas nunca usei
  - ☐ Nunca ouvi falar
- 

**8. Você se sente confortável para trabalhar com metodologias ágeis? \***

- ☐ Sim
- ☐ Em parte
- ☐ Não

**APÊNDICE D - QUESTIONÁRIO PARA VALIDAÇÃO DE CONHECIMENTO**



## Questionário - Feedback da Aplicação do Processo Proposto

---

### 1. Informe o seu curso de graduação \*

- ☐ Engenharia de software
  - ☐ Ciência da computação
- 

### 2. Informe qual a função exercida no projeto \*

- ☐ Líder técnico
  - ☐ Analista de requisitos
  - ☐ Analista de qualidade
  - ☐ Desenvolvedor
- 

### Planejamento e Requisitos

Responda cada afirmativa utilizando a escala de 1 a 5:

**1 – Discordo totalmente | 2 – Discordo | 3 – Neutro | 4 – Concordo | 5 – Concordo totalmente**

### 3. O refinamento do backlog melhorou o entendimento das funcionalidades a serem desenvolvidas. \*

☐ 1   ☐ 2   ☐ 3   ☐ 4   ☐ 5

### 4. A priorização dinâmica no quadro facilitou o ajuste de prioridades durante a Sprint. \*

☐ 1   ☐ 2   ☐ 3   ☐ 4   ☐ 5

### 5. As funcionalidades estavam bem descritas e sem ambiguidade. \*

☐ 1   ☐ 2   ☐ 3   ☐ 4   ☐ 5

---

### Execução e Fluxo de Trabalho

Responda cada afirmativa utilizando a mesma escala de 1 a 5.

### 6. A estrutura do quadro Kanban ajudou a visualizar e acompanhar o andamento das tarefas. \*

☐ 1   ☐ 2   ☐ 3   ☐ 4   ☐ 5

### 7. As reuniões diárias contribuíram para manter a equipe alinhada e resolver impedimentos rapidamente. \*

☐ 1   ☐ 2   ☐ 3   ☐ 4   ☐ 5

### 8. O fluxo de trabalho definido foi claro e fácil de seguir. \*

☐ 1   ☐ 2   ☐ 3   ☐ 4   ☐ 5

**9. A inspeção contínua antes da fase de testes ajudou a evitar retrabalho. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

---

### **Métricas e Visibilidade**

Escala de 1 a 5.

**10. O monitoramento do cycle time trouxe mais clareza sobre o tempo de desenvolvimento das tarefas. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**11. A limitação de WIP ajudou a evitar sobrecarga e aumentar o foco nas tarefas em andamento. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**12. O registro de bugs no quadro facilitou o controle e a resolução dos problemas encontrados. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

---

### **Resultados Percebidos**

Escala de 1 a 5.

**13. O novo processo contribuiu para melhorar a organização do time. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**14. Houve ganho de produtividade e agilidade na entrega das demandas. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**15. O processo ajudou a reduzir falhas ou retrabalho. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

---

### **Avaliação Geral**

Escala de 1 a 5.

**16. Estou satisfeito com o novo processo de desenvolvimento. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**17. Considero viável continuar utilizando esse modelo em projetos futuros. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

**18. O processo contribuiu positivamente para minha atuação na Sprint. \***

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

---

**19. O que funcionou bem durante a Sprint?**

---

**20. O que poderia ser melhorado no processo?**

---

**21. Outros comentários ou sugestões:**

---