

**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
CURSO DE ENGENHARIA DE PRODUÇÃO**

Arlen Ricardo Melgueiro Nunes

**MÉTODOS ÁGEIS APLICADOS NO GERENCIAMENTO DE PROJETOS DE UMA
STARTUP EM ESTÁGIO INICIAL**

**MANAUS
2025**

Arlen Ricardo Melgueiro Nunes

**MÉTODOS ÁGEIS APLICADOS NO GERENCIAMENTO DE PROJETOS DE UMA
STARTUP EM ESTÁGIO INICIAL**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Produção da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como parte dos requisitos para a obtenção do grau de Bacharel em Engenharia de Produção.

Orientador: Prof. MSc. Silvio Romero Adjar Marques.

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

N972m

Nunes, Arlen Ricardo Melgueiro

MÉTODOS ÁGEIS APLICADOS NO GERENCIAMENTO DE
PROJETOS DE UMA STARTUP EM ESTÁGIO INICIAL / Arlen

Ricardo Melgueiro Nunes. Manaus : [s.n], 2025.

47 f.: color.; 21.0 cm.

TCC - Graduação em Engenharia de Produção- Universidade do
Estado do Amazonas, Manaus, 2025.

Inclui Bibliografia.

Orientador: Marques, Silvio Romero Adjar.

1. Metodologias Ágeis. 2. Gestão de Projetos. 3. Lean Startup. 4.
Kanban. 5. Scrum. I. Marques, Silvio Romero Adjar (Orient.) II.
Universidade do Estado do Amazonas. III. Título

CDU(1997)658.5

ARLEN RICARDO MELGUEIRO NUNES

**MÉTODOS ÁGEIS APLICADOS NO GERENCIAMENTO DE PROJETOS DE UMA
STARTUP EM ESTÁGIO INICIAL**

Trabalho apresentado ao curso de Engenharia de Produção da Universidade do Estado do Amazonas, como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Produção.

Data de aprovação: Manaus (AM), 12 de novembro de 2025.

Banca examinadora:



Silvio Romero Adjar Marques, Prof. Me
Universidade do Estado do Amazonas



Cleto Cavalcante Souza Leal, Prof. Me
Universidade do Estado do Amazonas



Carly Pinheiro Trindade, Prof. Me
Universidade do Estado do Amazonas

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me conceder força, sabedoria e perseverança ao longo desta jornada.

À Nossa Senhora Auxiliadora, por sua intercessão constante e maternal, que me guiou com ternura nos momentos de dúvida e dificuldade.

À São José, modelo de trabalho, humildade e dedicação, por ser inspiração de coragem e silêncio fecundo diante dos desafios.

Ao meu orientador, Prof. MSc. Silvio Romero Adjar Marques, pela orientação precisa, paciência e incentivo constante, fundamentais para a realização deste trabalho.

À minha esposa Livia Reny Viana de Oliveira e filhas Alicia Rebeca de Oliveira Nunes e Alana Vitória de Oliveira Nunes, pelo apoio incondicional, amor e compreensão em todos os momentos, especialmente nos mais desafiadores.

Aos colegas, amigos e professores que contribuíram direta ou indiretamente com palavras de encorajamento, troca de ideias e apoio emocional.

À Universidade do Estado do Amazonas e à Escola Superior de Tecnologia, por proporcionarem o ambiente acadêmico e os recursos necessários para minha formação.

A todos que, de alguma forma, fizeram parte desta trajetória, minha sincera gratidão.

RESUMO

Este trabalho tem como objetivo propor uma metodologia híbrida de gestão de projetos baseada em métodos ágeis, voltada para startups em estágio inicial. A pesquisa foi desenvolvida a partir de um estudo de caso de uma empresa incubada na Universidade do Estado do Amazonas (UEA), que enfrenta desafios recorrentes relacionados à gestão e à entrega de projetos de software em ambientes de alta incerteza. A abordagem híbrida considera a aplicação integrada de metodologias como XP, Scrum, Kanban e Lean Startup, adaptadas às necessidades específicas das startups, que demandam flexibilidade, rapidez e inovação constante. A proposta visa melhorar a produtividade das equipes, a qualidade dos produtos entregues e o cumprimento dos prazos estabelecidos, por meio de práticas ágeis que favorecem a entrega contínua de valor e a validação incremental de soluções. A pesquisa busca validar a hipótese de que a adoção estruturada de métodos ágeis pode impactar positivamente o desempenho operacional e estratégico de startups, contribuindo para sua consolidação e competitividade no mercado.

Palavras-chave: Metodologias Ágeis, Gestão de Projetos, Lean Startup.

ABSTRACT

This work aims to propose a hybrid project management methodology based on agile methods, specifically designed for early-stage startups. The research was developed through a case study of a company incubated at the State University of Amazonas (UEA), which faces recurring challenges related to project management and software delivery in highly uncertain environments. The hybrid approach considers the integrated application of methodologies such as XP, Scrum, Kanban, and Lean Startup, adapted to the specific needs of startups that require flexibility, speed, and constant innovation. The proposal seeks to improve team productivity, the quality of delivered products, and the fulfillment of established deadlines through agile practices that promote continuous value delivery and incremental solution validation. The research aims to validate the hypothesis that the structured adoption of agile methods can positively impact the operational and strategic performance of startups, contributing to their consolidation and competitiveness in the market.

Keywords: Agile Methodologies, Project Management, Lean Startup.

LISTA DE ILUSTRAÇÕES

Figura 1 – Formato de História de Usuário.....	15
Figura 2 – Deck Fibonacci.....	15
Figura 3 – Quadro Kanban comumente utilizado no desenvolvimento de software	16
Figura 4 – Cartas de Planning Poker.....	17
Figura 5 – Kanban via Trello.....	17

LISTA DE TABELAS

Tabela 1 – Práticas de gestão metodologia Scrum.....	13
Tabela 2 - Componentes e Cerimônias do framework de desenvolvimento....	14
Tabela 3 – Conceitos e Práticas das metodologias propostas.....	16
Tabela 4 - Tabela Analítica de Resultados.....	17

SUMÁRIO

1. INTRODUÇÃO	11
1.1 TEMA.....	12
1.2 PROBLEMA DE PESQUISA.....	12
1.3 HIPÓTESE.....	13
1.4 OBJETIVOS.....	14
1.4.1 Objetivo Geral	14
1.4.2 Objetivos Específicos	14
1.5 JUSTIFICATIVA	14
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 SCRUM.....	19
2.2 XP (EXTREME PROGRAMMING).....	21
2.3 LEAN SOFTWARE DEVELOPMENT	23
2.4 HISTÓRIAS DE USUÁRIO	26
2.5 KANBAN	28
3 MATERIAIS E MÉTODOS	30
3.1 APRESENTAÇÃO DO CASO.....	30
3.2 DESCRIÇÃO DO CONTEXTO ESTUDADO.....	32
3.3 APLICAÇÃO DOS CONCEITOS ÁGEIS NA CONSTRUÇÃO DA METODOLOGIA HÍBRIDA. 33	
3.4 ANÁLISE DA APLICAÇÃO DA METODOLOGIA HÍBRIDA.....	38
4 RESULTADOS E DISCUSSÃO	43
5 CONCLUSÃO	44

1. INTRODUÇÃO

O mercado de desenvolvimento de software tem se consolidado como um dos pilares da transformação digital em diversos setores da economia brasileira (Mercado Brasileiro de Software: Panorama e Tendências, 2025). Com a crescente demanda por soluções tecnológicas ágeis, escaláveis e personalizadas, as startups têm ganhado destaque como agentes de inovação, atuando em ambientes dinâmicos e de alta incerteza. Essas empresas emergentes, muitas vezes incubadas em instituições de ensino e pesquisa, enfrentam desafios únicos relacionados à gestão de projetos, à alocação de recursos e à adaptação constante às mudanças do mercado. Nesse contexto, metodologias ágeis são alternativas eficazes para lidar com tais desafios, promovendo entregas rápidas, colaboração contínua e foco no valor gerado ao cliente.

Nos últimos anos, as metodologias ágeis têm ganhado importância no cenário de desenvolvimento de software, pois quando implementadas em seus projetos permitem atingir melhores resultados (Agile Project Delivery Confidence, 2017) e maior satisfação dos clientes (Wouter Aghina, *et al*, 2025). A principal diferença entre estas e as metodologias tradicionais está nos princípios utilizados para atingir os objetivos do projeto. O planejamento detalhado é feito apenas para a fase atual do projeto, e os planos para fases futuras são considerados rascunhos, que podem ser alterados de acordo com o aprendizado da equipe e a evolução do projeto (Sato, 2007). Durante o desenvolvimento de projetos ágeis, é comum que os requisitos definidos inicialmente pelo usuário mudem. Não raramente as mudanças ocorrem devido à evolução do negócio ao qual o projeto está atrelado, e o custo para alteração de um requisito, após a entrega de um software, é considerado muito alto.

Um dos princípios das metodologias ágeis é aceitar as mudanças, mesmo que isto ocorra em estágios mais avançados do desenvolvimento, pois tais mudanças podem trazer vantagem competitiva para o cliente (Sato, 2007). Além da aceitação de mudanças, as metodologias ágeis seguem outros princípios que trazem mais dinamismo para o processo de desenvolvimento, incluindo a proximidade com o cliente, priorizando sua satisfação, e a motivação da equipe, para que desempenhem seus papéis da melhor forma possível.

A maioria dos processos de desenvolvimento de projetos de software foi concebida para empresas já bem estruturadas. Porém, com o crescimento do número de startups ao redor do mundo, processos mais apropriados para o perfil dessas

empresas tiveram que ser pensados. A adoção de metodologias ágeis em startups pode contribuir para sua formação e crescimento, por se tratar de métodos adaptativos. Para Gitahy (2010), startups buscam concretizar um modelo de negócios em um ambiente de incertezas. Isso pode significar que mudanças sejam necessárias para que se alcance um modelo de negócios mais sólido e eficaz.

Este trabalho apresenta um estudo sobre metodologias ágeis, com enfoque em quatro principais metodologias: *XP*, *Scrum*, *Kanban*, *Lean Startup*, trazendo-os para uma realidade de uma startup que enfrenta problemas de gestão ocasionando a não conclusão de etapas do projeto no período determinado e com conhecimentos limitados da metodologia *Lean Startup*, cuja aplicabilidade é valiosa para a atual situação da startup objeto deste estudo. Com base nas práticas das metodologias mencionadas, será proposta uma metodologia híbrida adaptada para o ambiente de startup com fins de analisar sua utilização em uma empresa específica, que será chamada de Startup X, e não terá seu nome revelado por questões de privacidade.

1.1 TEMA

Este trabalho explora como as metodologias ágeis, especialmente *XP*, *Scrum*, *Kanban* e *Lean Startup*, impactam o desenvolvimento do projeto de software em ambientes de incerteza. Será realizado um estudo com uma abordagem objetiva, estabelecendo metas específicas, mensuráveis, alcançáveis, relevantes e com prazo definido, integrando elementos de diferentes metodologias para propor a melhoria da gestão de projetos em uma startup que se encontra na fase inicial de desenvolvimento, adotando modelos e práticas de metodologias ágeis possam acelerar o progresso do projeto.

1.2 PROBLEMA DE PESQUISA

Metodologias ágeis têm sido amplamente adotadas no desenvolvimento de software devido à sua adaptabilidade e à capacidade de responder a mudanças rápidas. No entanto, sua implementação em startups apresenta desafios específicos, como a limitação de recursos, sejam financeiros, sejam materiais e/ou de recursos humanos, havendo a necessidade de experimentação constante e a busca por um modelo de negócios sustentável.

No contexto abordado por este estudo, a startup X vem enfrentando problemas de entrega de resultados por ineficiência de gerenciamento efetivo, e falta de boas práticas de gestão.

Visando elucidar essa relação, o presente estudo contempla um estudo de caso na startup X, aliada a uma combinação de metodologias ágeis, propondo uma modelo de práticas adaptado para o ambiente de startup, visando servir como papel fundamental aos processos de inovação. Sendo assim, a questão problema é "Como a combinação de metodologias ágeis, aliada à gestão de projetos, pode impactar positivamente o desenvolvimento de projetos e o crescimento de uma startup?"

1.3 HIPÓTESE

A Startup X enfrenta dificuldades no desenvolvimento de projetos, resultando em atrasos nas entregas e longo lead time para a criação de produtos. Esses problemas decorrem da ausência de práticas estruturadas por parte da gestão, o que compromete a entrega das etapas previstas no cronograma. Diante desse cenário, a hipótese é que a implementação de uma metodologia híbrida de desenvolvimento ágil contribuirá para:

- Aumentar a produtividade da equipe, por meio de maior organização e foco nas entregas;
- Elevar a qualidade do software, com ciclos de desenvolvimento mais curtos e feedback contínuo;
- Garantir o cumprimento dos prazos estabelecidos, por meio de maior flexibilidade e controle das etapas do projeto.

A adoção dessa abordagem visa transformar o processo de desenvolvimento da Startup X, tornando-o mais eficiente, inovador e alinhado às expectativas dos clientes.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Apresentar uma proposta de gestão de projetos baseada em metodologias ágeis, com foco na evolução e desenvolvimento de produto para a startup X que se encontra em estágio inicial de desenvolvimento.

1.4.2 Objetivos Específicos

- Realizar pesquisa bibliográfica sobre conceitos e aplicações das metodologias ágeis na gestão de projetos;
- Analisar o conceito e as características da abordagem Lean Software Development;
- Estruturar os conceitos pesquisados em uma proposta aplicável ao contexto da Startup X.

1.5 JUSTIFICATIVA

A adoção de metodologias ágeis tem se mostrado um fator decisivo para o sucesso de projetos em ambientes de elevada incerteza, como é o caso de startups. De acordo com a PwC (Agile Project Delivery Confidence, 2017), projetos geridos com abordagem ágil são 28% mais bem-sucedidos do que aqueles conduzidos por métodos tradicionais. Essa taxa de sucesso é mensurada por parâmetros como entrega contínua de valor ao cliente, maior velocidade de lançamento no mercado, redução de riscos por meio de ciclos curtos e feedback constante, aumento da qualidade do produto entregue e maior alinhamento entre os objetivos estratégicos e os resultados obtidos.

Complementando essa perspectiva, o estudo *The Impact of Agility: How to Shape Your Organization to Compete* da McKinsey & Company (2021) reforça que empresas que implementam metodologias ágeis de forma estruturada apresentam ganhos expressivos:

- 30% de aumento na satisfação dos clientes;

- 30% de melhoria na performance operacional;
- Cinco a dez vezes mais rapidez na entrega de soluções;
- Três vezes mais chances de se tornarem líderes de mercado, em comparação com organizações que não passaram por transformações ágeis.

Além disso, a McKinsey destaca que transformações ágeis bem-sucedidas não se limitam à adoção de práticas em equipes isoladas, mas envolvem uma reestruturação profunda do modelo operacional — incluindo estratégia, estrutura, processos, pessoas e tecnologia. Essa abordagem sistêmica permite que a organização funcione como uma rede de equipes de alto desempenho, com foco claro em resultados de negócio e capacidade de adaptação contínua.

Portanto, diante dos desafios enfrentados pela Startup X, a aplicação de uma metodologia híbrida de desenvolvimento ágil se apresenta como uma solução estratégica. Ela não apenas permite potencializar a produtividade e a qualidade do software entregue, como também fortalece a capacidade da organização de responder rapidamente às mudanças do mercado, atendendo os prazos estipulados e gerando valor real para os stakeholders.

2 FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento de software não é uma tarefa simples. Ao se iniciar um projeto desta natureza, deve-se considerar que ao final de todos os processos seja gerado um produto de qualidade, que atenda amplamente aos interesses e necessidades do cliente, e que seja entregue dentro do prazo e do orçamento previstos. Atingir estas exigências é o principal desafio do processo de desenvolvimento de software.

Projetos de criação de empresas de software envolve desafios complexos, exigindo que, ao final do processo, o produto entregue atenda às necessidades do cliente, tenha qualidade elevada e respeite os prazos e custos previstos. Durante as décadas de 1980 e 1990, predominava a crença de que a melhor abordagem para a construção de software envolvia um planejamento detalhado, com segurança formalizada, além de um rigoroso controle do processo de desenvolvimento, (Sommerville, 2011). Esse método, embora adequado para grandes projetos com múltiplas equipes ou sistemas críticos, demanda tempo significativo para planejamento, projeto e documentação.

No entanto, essa abordagem estruturada nem sempre resultava em eficiência para sistemas de pequeno e médio porte, gerando insatisfação entre os profissionais da área, o que levou ao surgimento das metodologias ágeis na década de 1990 (Sommerville, 2011). Em 2001, especialistas em desenvolvimento de software se reuniram para discutir alternativas às metodologias tradicionais. Embora reconhecessem a importância da modelagem e documentação, enfatizavam a necessidade de equilíbrio, evitando processos excessivamente burocráticos. Além disso, defendiam o planejamento, mas com compreensão dos limites da previsibilidade (Dias, 2010).

Ludvig e Reinert (2007), citam que as abordagens ágeis surgiram como uma reação às metodologias pesadas (Fowler, 2005) e tiveram como principal motivação criar alternativas para o Modelo Tradicional em Cascata (Hilman, 2004). Carvalho, Abrantes e Cameira (2011) mostram que estas metodologias focam nos indivíduos e na interação entre eles, no funcionamento do software, na colaboração com o cliente e na resposta rápida às mudanças. Isto é alcançado através de ciclos iterativos de desenvolvimento de sistemas com objetivos de curto prazo, tornando possível a adaptação dos requisitos até o momento em que entram em fase de desenvolvimento.

Segundo Fowler (2005), as metodologias ágeis de desenvolvimento são um conjunto de práticas e métodos criados e desenvolvidos ao longo das últimas duas décadas, que têm por objetivo tornar o desenvolvimento de software rápido, com custo controlável e melhorar a qualidade do software. Para ele, a maior diferença entre as metodologias tradicionais e as metodologias ágeis são as premissas de que as metodologias ágeis são adaptativas ao invés de predeterminantes. Metodologias tradicionais tentam prever detalhes do processo de desenvolvimento no início do projeto, e acabam tendo alguns problemas quando são necessárias mudanças no projeto, como atrasos na entrega do projeto.

Nas metodologias ágeis, a premissa é de que sempre haverá mudanças no projeto que não podem ser previstas em seu início, e essas mudanças são bem-vindas, estando todos os envolvidos no projeto preparados para elas. E as metodologias ágeis são orientadas a pessoas e não a processos. Metodologias tradicionais definem um processo que irá funcionar bem, independente de quem os utilize. Portanto, o papel do processo é dar suporte à equipe de desenvolvimento e seu trabalho.

O movimento originado com a criação da abordagem ágil tinha como premissa central um modelo de desenvolvimento pautado na agilidade, flexibilidade e comunicação eficaz, possibilitando a entrega contínua de novos produtos e serviços com valor agregado em períodos reduzidos. De acordo com Dias (2010), a agilidade não deve ser confundida com ausência de estrutura, mas entendida como a capacidade de adaptação e resposta às mudanças.

Como resultado dessa percepção foi criada a Aliança Ágil e estabelecido o Manifesto Ágil, que é baseado em quatro valores:

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

Ou seja, embora haja valor nos itens à direita, nós valorizamos mais os itens à esquerda. (Beck *et al.*, 2001)

Para auxiliar a compreensão destes valores, foram definidos doze princípios para a prática de metodologias ágeis.

- a) Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
- b) Mudanças nos requisitos são bem-vindas mesmo diante no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
- c) Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- d) Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
- e) Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
- f) O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.
- g) Software funcionando é a medida primária de progresso.
- h) Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
- i) Contínua atenção à excelência técnica e bom design aumenta a agilidade
- j) Simplicidade - arte de maximizar a quantidade de trabalho não realizado – é essencial.
- k) As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.
- l) Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Os princípios básicos de métodos ágeis compreendem honestidade ao código de trabalho, eficácia das pessoas que trabalham em conjunto e foco no trabalho em equipe. Assim, o grupo de desenvolvimento, incluindo desenvolvedores e representantes do cliente, deve ser bem-informado, competente e autorizado a considerar o eventual ajuste das necessidades emergentes durante o processo de ciclo de vida do desenvolvimento. Isto significa que os participantes estão preparados para fazer mudanças, e que também os contratos existentes são formados com as ferramentas que suportam e permitem que essas melhorias sejam feitas (LEITÃO, 2010).

Ludvig e Reinert (2007) afirmam que o Manifesto Ágil não é contra os modelos adotados pela abordagem tradicional. Os métodos ágeis são fundamentados no desenvolvimento incremental, incluem técnicas utilizadas na Engenharia de Software, porém não seguem o padrão proposto pelas metodologias tradicionais. Portanto, não rejeitam os processos e ferramentas, a documentação, a negociação de contratos ou o planejamento, apenas mostram que eles têm importância secundária comparados com os indivíduos e as iterações, como o software estar executável, com a colaboração e as respostas rápidas a mudanças e alterações.

2.1 SCRUM

O Scrum é uma metodologia ágil de desenvolvimento de software, criado na década de 1990 por Jeff Sutherland, com base em conceitos tradicionais da engenharia de produção como Lean e a Teoria das Restrições Explorada por Takeuchi e Nonaka (1986) no artigo *“The New Product Development Game”*, no qual descrevem as vantagens da utilização de times pequenos, multidisciplinares e autogerenciáveis no desenvolvimento de produtos. O objetivo do Scrum é entregar a maior qualidade de software possível por meio de uma série de pequenos intervalos de tempo fixo, chamados Sprints, que tipicamente duram menos de um mês (Sutherland et al., 2000). Ou seja, o objetivo maior da sprint é entregar o máximo valor de negócio possível, no menor tempo. Schwaber e Sutherland (2011), afirmam que o Scrum é fundamentado nas teorias empíricas de controle de processo. O empirismo estabelece que o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido. Ele se apoia em três pilares:

1. Transparência - Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados;
2. Inspeção - Os usuários Scrum devem, frequentemente, inspecionar os artefatos Scrum e o progresso em direção ao objetivo, para detectar indesejáveis variações. A inspeção não pode sobrepor a execução das atividades;
3. Adaptação - Se for verificado que algum aspecto do processo se desviou dos limites aceitáveis e que o produto será inaceitável, o processo deve ser reajustado para produzir o que era esperado. E isso deve ser feito o mais breve possível para minimizar os desvios.

Há quatro cerimônias principais relacionados ao Scrum que apoiam o desenvolvimento do projeto. São eles:

a) Reunião de planejamento da Sprint: É uma etapa do framework Scrum realizada no início de cada sprint. Nela, o Product Owner apresenta os itens prioritários do backlog, e o time de desenvolvimento define quais tarefas serão realizadas no ciclo. A equipe estabelece uma meta clara para a sprint e planeja como executar o trabalho. O resultado é um sprint backlog definido e todos os membros alinhados com os objetivos.

b) Daily Scrum (Daily Meeting): É uma reunião diária e rápida, realizada pelo time de desenvolvimento dentro do framework Scrum. Seu objetivo é promover alinhamento entre os membros da equipe sobre o progresso da sprint. Cada participante compartilha o que fez no dia anterior, o que pretende fazer no dia atual e se há algum impedimento. A reunião dura cerca de 15 minutos e ajuda a manter o foco, identificar problemas rapidamente e garantir a colaboração contínua.

c) Reunião da revisão da Sprint: É realizada ao final de cada sprint no framework Scrum. Seu objetivo é apresentar o que foi desenvolvido à equipe e aos stakeholders, avaliando se a meta da sprint foi atingida. O time demonstra os incrementos concluídos e recebe feedback, promovendo transparência e colaboração. Essa reunião também ajuda a ajustar o backlog do produto com base nas novas informações obtidas.

d) Retrospectiva da Sprint: É uma reunião realizada ao final de cada sprint no framework Scrum. Seu objetivo é permitir que o time reflita sobre o processo de trabalho, identifique pontos positivos, dificuldades e oportunidades de melhoria. A equipe discute o que funcionou bem, o que pode ser ajustado e define ações concretas para aprimorar a próxima sprint. Essa prática fortalece a colaboração, a transparência e a melhoria contínua.

O Scrum possui três papéis bem definidos que são apresentados na Tabela 1:

QUADRO 1: PRÁTICAS DE GESTÃO METODOLOGIA SCRUM

Termo	Definição
Scrum Master	É o responsável por garantir que tudo vai ocorrer de acordo com as regras do Scrum (práticas, valores) e que o projeto tenha o progresso esperado. Ele também é responsável por eliminar ou mitigar qualquer impedimento que possa atrapalhar a produtividade da equipe.

Product Owner	É o responsável por gerenciar as atividades que serão desenvolvidas (Product Backlog). Ele toma as decisões quanto à priorização das atividades e participa da estimativa do esforço de desenvolvimento.
Scrum Team	É a equipe de desenvolvimento propriamente dita

FONTE: ELABORADO PELO AUTOR (2025), BASEADO EM VIEIRA (2014)

Descritos na Tabela 2 estão ações que se referem aos artefatos (listas de trabalho) utilizados no gerenciamento do scrum, de forma iterativa e incremental.

QUADRO 2: COMPONENTES DO FRAMEWORK DE DESENVOLVIMENTO

Ação	Definição
Backlog do Produto	É a definição de tudo que é necessário para o produto final, baseado no conhecimento atual. É uma lista de prioridades que é constantemente atualizada pelo Product Owner, como: História de usuário, Requisitos Funcionais, Requisitos não funcionais, correções de bug, melhorias técnicas ou refatoração e Requisitos de Negócio.
Effort Estimation	É o momento em que o time, juntamente com o Product Owner e o Scrum Master, define o tempo estimado para cada item do Product Backlog e verifica-se as informações existentes sobre o item são suficientes.
Sprint	É o período de desenvolvimento, de tempo predeterminado, para as atividades que foram selecionadas para aquele incremento de valor do produto.
Sprint Backlog	É a lista de atividades que foram escolhidas para ser desenvolvida na sprint. Esta lista não muda até que termine a Sprint. Quem deve garantir que isso aconteça é o Scrum Master.

FONTE: ELABORADO PELO AUTOR (2025), BASEADO EM VIEIRA (2014).

2.2 XP (EXTREME PROGRAMMING)

XP é uma metodologia de desenvolvimento de software que combina rapidez, produtividade e qualidade de forma simples, e que atende às necessidades do cliente. É uma metodologia voltada para o desenvolvimento em que os requisitos se modificam constantemente. Ela busca gerar o máximo de valor possível para o cliente, em um curto espaço de tempo. E durante esse espaço de tempo, o cliente tem a possibilidade de analisar o produto recebido e verificar se é realmente o desejado.

Segundo Beck (2004), a XP é uma metodologia ágil para equipes médias e pequenas, onde os requisitos para o desenvolvimento de software são vagos e em constante mudança. Extreme Programming traz em sua base valores e práticas que

sempre procuram garantir ao cliente versatilidade e satisfação com o produto final. Os valores do XP são definidos como:

- a) Comunicação: É um ponto extremamente importante para o bom andamento de um projeto, pois, a conversação entre o cliente e a equipe tem que ser bem clara e objetiva;
- b) Simplicidade: tudo no XP deve ser feito da forma mais simples possível. O objetivo é fazer as coisas da forma mais simples que. Dessa forma o cliente terá a funcionalidade rapidamente da forma desejada;
- c) Feedback: o cliente está sempre aprendendo com o produto que está recebendo durante o projeto. Com isto, ele consegue tomar decisões de priorização de atividades, consegue verificar se o que está sendo feito é realmente o que ele necessitava;
- d) Coragem: algumas práticas no XP, tais como testes unitários, integração contínua, programação em par entre outras, aumentam a confiança da equipe e ajudam a ter coragem para melhorar o código que está funcionando e torná-lo mais simples (refatorar).

O Extreme Programming possui 12 práticas que a tornam uma metodologia mais próxima do cliente, e permitem que a equipe avance com celeridade na resolução dos problemas.

- a) Planning Game: Reunião entre clientes e desenvolvedores para definir e planejar as atividades da iteração;
- b) Cliente Presente (On-site Customer): O cliente deve estar disponível para esclarecer dúvidas e orientar o desenvolvimento;
- c) Programação em Duplas (Pair Programming): Duas pessoas trabalham juntas no mesmo código, promovendo qualidade e troca de conhecimento;
- d) Código Coletivo (Collective Code Ownership): Todo o código pertence à equipe; qualquer membro pode modificar qualquer parte;
- e) Metáfora: Uso de analogias para facilitar a compreensão do sistema por todos, inclusive pelo cliente;
- f) Padrão de Código: A equipe estabelece e segue um conjunto de padrões para nomear variáveis, classes e métodos, e para organizar o código. Isso torna o sistema mais homogêneo e compreensível, facilitando a manutenção e a posse coletiva;

- g) Releases Curtos: Entregas frequentes de funcionalidades que geram valor ao cliente com rapidez;
- h) Refatoração: Melhorias contínuas no código de forma a torná-lo mais simples e eficiente.
- i) Integração Contínua: Integração frequente das partes do sistema, com testes constantes para garantir estabilidade;
- j) Design Simples: O sistema deve ser projetado da forma mais simples possível, atendendo apenas às necessidades atuais;
- k) Desenvolvimento Orientado a Testes: Criação de testes durante o desenvolvimento, para a garantia de que o código funcione corretamente;
- l) Semana de 40 Horas (Ritmo Sustentável): A prática preza por uma jornada de trabalho de 40 horas semanais, evitando horas extras. O objetivo é manter a equipe, produtiva e trabalhando em um ritmo constante a longo prazo, garantindo a sustentabilidade do projeto

2.3 LEAN SOFTWARE DEVELOPMENT

É uma aplicação dos princípios e práticas de Lean no contexto de desenvolvimento de software.

Lean é uma metodologia de produção que tem foco em eliminar os desperdícios, aumentar a velocidade de processos e melhorar a qualidade do processo ou do produto. Ele considera desperdício qualquer gasto de recurso com outro objetivo que não seja a criação de valor para o cliente final, o que deixa o processo sem excessos e com o máximo de atividades que gerem valor, deixando o processo lean, ou em outras palavras: enxuto.

Pantalião (2009), trouxe em seu texto, “O que é Lean?” Algumas definições bem esclarecedoras:

- a) Allan Shalloway afirma: “Lean é uma abordagem para entregar valor mais rapidamente para seus clientes, focando em melhorar o workflow dos produtos sendo entregues. Em particular, times devem sempre reduzir atrasos, que estão ligados a desperdício e baixa qualidade”;
- b) Mary Poppendieck complementa, citando a opinião de John Shook sobre o assunto: “Como sabemos, muito e se não a maior parte, se não quase tudo do

"Lean" é essencialmente procurarmos meios de executar o P-D-C-A em tudo que fazemos";

- c) Kent Beck resume muito bem sua opinião: "Ao aplicar o desenvolvimento Lean, eliminamos tempo e esforço desperdiçado. O que sobra é produtividade. Quando isto tiver sido feito, repetimos a aplicação e desta forma, o ciclo de melhoria nunca termina";
- d) Mary Poppendieck comenta novamente e faz algumas observações na mesma linha de Kent Beck: "Lean é sobre sempre melhorar o que você é agora através de testes e métricas dos resultados. Então, é sobre nunca estar confortável com o status quo, e sempre aprender como melhorar, utilizando o método científico." Os usuários de um software irão perceber a integridade de um sistema quando ele resolve o seu problema de uma forma fácil de usar e rentável. Não importa se o problema é mal compreendido, se existiram mudanças ao longo do tempo, ou é dependente de fatores externos, um sistema com integridade percebida o que resolve o problema de forma eficaz (Poppendieck; Poppendieck, 2003).

Para a implementação do Lean, é necessária uma mudança de cultura na organização. Não basta trazer uma nova série de regras e imaginar que automaticamente a organização se tornará enxuta. Esta mudança na forma de pensar da equipe é certamente o maior desafio na execução desta iniciativa do Lean aplicados ao Desenvolvimento de Software:

- a) Elimine os desperdícios "Desperdício é tudo aquilo que não agrega valor ao cliente", Taiichi Ohno. Devemos eliminar toda e qualquer coisa que não traga valor ao cliente. É muito difícil entender o conceito real de valor e identificá-lo. Alguns exemplos de desperdício são: funcionalidades extras; documentação produzida apenas para "regularizar" o processo; funcionalidades incompletas ou com defeito. É essencial identificar a necessidade do cliente e entregar a funcionalidade da forma mais rápida possível e com qualidade.
- b) Amplifique o aprendizado; desenvolvimento de software pode ser considerado um exercício de constantes descobertas. Não se espera que um desenvolvedor consiga gerar o melhor código de primeira. Ele vai "experimentando" e aprendendo coisas novas, e conforme ganha experiência vai surgindo um software melhor.

- c) Decida o mais tarde possível; A prática de adiar a decisão garante que esta não seja tomada em um momento de incerteza, adiando-a, até que haja uma maturidade maior.
- d) Entregue o mais rápido possível: “A moral da história é que devemos encontrar uma maneira de entregar software tão rápido que nossos clientes não tenham tempo de mudar de ideia”. (Mary Poppendieck, 2003). O feedback é um item essencial para o Lean. No desenvolvimento, há um ciclo de descoberta que é fundamental para a aprendizagem: histórias, implementação, feedback e melhorias. Quanto menor esse ciclo, mais poderá ser aprendido. Após a solicitação do cliente, deve haver um esforço para entregar o software em funcionamento o mais rápido possível. A rapidez deste ciclo faz com que o cliente tenha o que necessita o quanto antes. Em uma organização madura em desenvolvimento de software, tudo isso acontece em um fluxo rápido, em resposta a uma necessidade dos clientes.
- e) Dê autonomia à equipe: Um dos doze princípios do manifesto ágil é "Construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho." Uma equipe deve ser suficientemente auto-organizada e multidisciplinar, com a capacidade de tomar decisões, assumir compromissos, riscos, ir atrás de seus próprios objetivos.
- f) Visualize o todo: O desenvolvimento de sistemas focando o todo visa diminuir problemas de integridade. Uma visão mais complexa faz com que o equilíbrio entre todas as particularidades de um sistema seja mantido e haja um melhor aproveitamento.

Gitahy (2015) aborda um importante conceito utilizado na metodologia Lean, o MVP (Minimum Viable Product ou Produto Mínimo Viável), que é um marco importante no ciclo de vida de um empreendimento. Para melhor explicar os conceitos do MVP, explicamos cada letra: (M) Minimum: o menor tamanho possível, que possa ser entregue no menor tempo possível. (V) Viable: uma proposição de valor importante o suficiente para que seu principal cliente adote esse produto, se possível gerando receita. (P) Product: são funcionalidades encaixadas em uma entrega que se assemelhe a um produto coeso e útil. (Gitahy, 2015). Integrar o entendimento entre “mínimo”, “produto” e “viável” exige muito esforço para encontrar o ponto de equilíbrio entre algo que o cliente dê valor, consumindo poucos recursos e que se assemelhe a

um produto. Por isso, a tarefa de encontrar um MVP não é simples, ainda mais se tratando de um negócio inovador. Normalmente isso só ocorrerá após várias iterações, tentativas e erros, protótipos testados com usuários e muito esforço (Gitahy, 2015).

2.4 HISTÓRIAS DE USUÁRIO

Histórias de usuário nasceram com o propósito expressar os requisitos das partes interessadas. Com o passar do tempo, muito impulsionado pela expansão do framework Scrum, o Product Owner tornou-se a pessoa principal a escrever essas histórias e organizá-las em um backlog de produto. Entretanto, qualquer um pode (e deve) escrever histórias de usuário que expressem suas dores e necessidades.

Aguiar e Caroli, (2024) afirmam que cada história é a apenas um “lembrete” da necessidade do cliente, ou seja, ela informa qual o requisito fornecido pelo cliente, sendo este documentado para ajudar na construção do produto.

História de usuário é um formato textual com uma descrição concisa de um requisito que busca responder três indagações básicas: Quem? O que? e por quê? Das seguintes formas (Figura 1):

Como: Perfil do usuário / Personas;

Posso: Função específica do produto;

Para: Valor do Negócio;

O critério de aceite é um roteiro textual que descreve como testar uma história de usuário, conforme critérios definidos para aceitar uma história como pronta. A história de usuário terá alguns critérios de aceite, como:

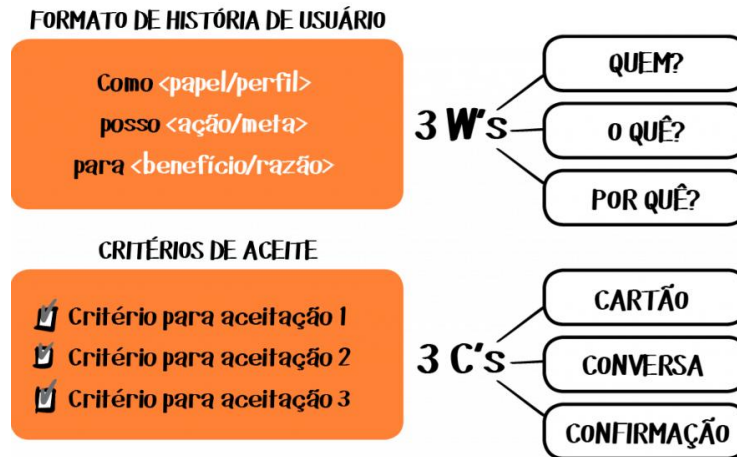
Dado que: Cenário inicial;

Quando: Ação realizada;

Então: Resultado esperado;

Os critérios de aceite determinam quando uma *história de usuário* está concluída, apoiando o trabalho do Product Owner, que realiza o aceite da história ao comprovar que todos os critérios foram atendidos.

FIGURA 1: FORMATO DE HISTÓRIA DE USUÁRIO



FONTES: PRODUCT BACKLOG BUILDING (PBB) (2024)

Um exemplo de aplicação do método:

Como: consumidor,
Posso: adicionar produtos ao carrinho,
Para: organizar minha compra antes de finalizar o pedido.

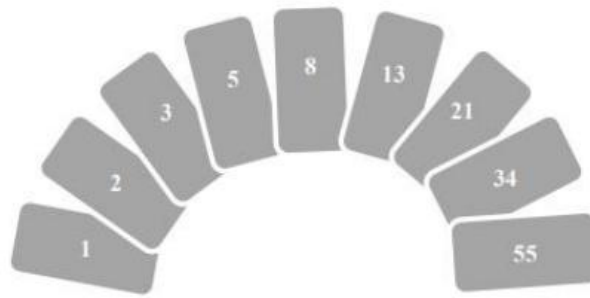
- Critérios de Aceite:

Dado que: estou conectado na minha conta,
Quando: clico no botão "Adicionar ao carrinho",
Então: o produto deve aparecer no carrinho com a quantidade e preço corretos.

Estimar o esforço de tarefas de forma colaborativa o mais próximo à realidade é de grande importância, pois, o tempo necessário para realizar o desenvolvimento implica no custo, ou seja, a estimativa estabelece o custo de desenvolvimento do projeto. A técnica mais popular para metodologias ágeis é a Planning Poker (Munialo; Muketha, 2016).

Para quantificar uma tarefa normalmente é utilizado um padrão de story points que representam o esforço para realizar a tarefa de forma quantitativa. A Figura 2 mostra um exemplo de escala muito utilizada, em que os valores obedecem a sequência de Fibonacci. Cada integrante da equipe determina uma quantidade de story points para uma tarefa de forma individual; quando todos tiverem definidos seus votos, todos os votos são revelados. Caso haja alguma diferença de opinião ocorre uma discussão entre as partes discordantes para definir um denominador comum. Esse processo é realizado para cada tarefa, até todas as histórias de usuário serem estimadas (Mahinic; Hovelja, 2012)

FIGURA 2: DECK FIBONACCI



FONTE:(MALLIDI; SHARMA, 2021)

As cartas do baralho têm números. Um baralho típico apresenta cartas que mostram a sequência de Fibonacci, incluindo o zero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89; outros baralhos usam progressões semelhantes com uma proporção fixa entre cada valor, como 1, 2, 4, 8, etc.

O Planning Poker deve forçar as pessoas a pensarem de forma independente e a proporem seus números simultaneamente. Isso é conseguido exigindo que todos os participantes mostrem suas cartas ao mesmo tempo.

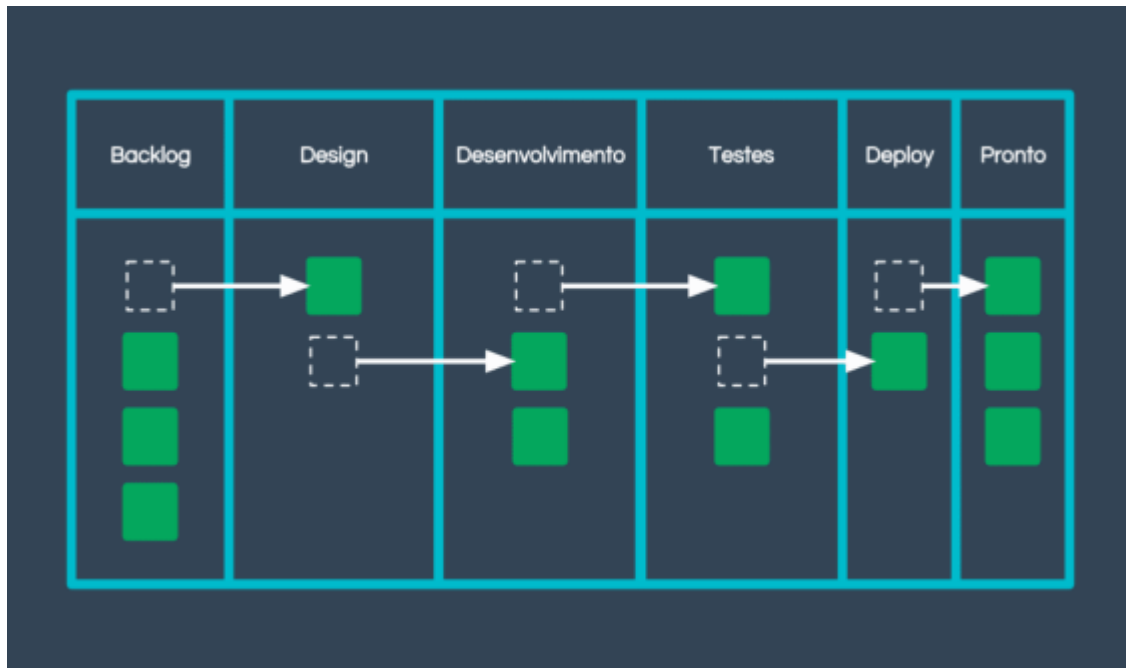
2.5 KANBAN

O Sistema Toyota de Produção desenvolveu uma ferramenta chamada *Kanban* no esforço necessário para realizar as atividades de construção das funcionalidades do produto para coordenar o fluxo de produtos visando o controle e nivelamento da produção e à minimização dos estoques intermediários e finais (OHNO, 1997). Taiichi Ohno extraiu sua ideia dos supermercados, onde a mercadoria só é resposta quando é retirada pelo cliente observando um modelo que assegure que seja produzida somente a quantidade de itens necessários por meio, do sistema puxado de produção.

O Kanban é definido como um sistema de controle visual simplificado, que é aplicado no chão de fábrica em empresas com produção repetitiva, servindo como instrumento para a melhoria contínua dos sistemas de produção (Shingo, 1996). A aplicação da abordagem Kanban à área de desenvolvimento de Software oferece uma sistemática flexível e transparente para gerenciamento de projetos de software, promovendo a colaboração, a eficiência e a melhoria contínua. Cada ciclo passará a ser mais curto, dando mais agilidade para a entrega de recursos ao usuário.

O método consiste no controle visual do fluxo de trabalho, através da composição exemplificada na figura 3, contendo o uso de cartões em um quadro, que pode ser físico ou digital, dividido em colunas que servem para representar as tarefas em diferentes fases de execução.

FIGURA 3: QUADRO KANBAN COMUMENTE UTILIZADO NO DESENVOLVIMENTO DE SOFTWARE



FONTE: MÉTODO KANBAN: UM GUIA (QUASE) COMPLETO (TARGETTEAL.COM) (2025)

O *Kanban* pode ser utilizado para auxiliar as equipes na organização de suas tarefas, garantindo que nenhuma atividade seja esquecida e que todos os clientes sejam atendidos de maneira ágil e eficiente. Funciona como um assistente pessoal, porém voltado para toda a equipe.

3 MATERIAIS E MÉTODOS

A realização de uma pesquisa científica exige, de forma imprescindível, a definição clara de um procedimento metodológico. Segundo Fiorese, citado por Gomes e Araújo (2005), a metodologia consiste em um conjunto de métodos que viabilizam a elaboração de estratégias para alcançar objetivos específicos. Nesse sentido, Lakatos e Marconi (2021) destacam que a metodologia representa o conjunto de etapas e técnicas que orientam a pesquisa, sendo responsável por organizar e sistematizar a coleta e a análise dos dados, elementos essenciais para a construção do conhecimento científico.

Assim, a metodologia pode ser compreendida como a aplicação de métodos que asseguram a validade e a confiabilidade dos dados, permitindo ao pesquisador alcançar conclusões fundamentadas e objetivas. Para que o conhecimento produzido seja considerado científico, é necessário identificar os processos mentais e técnicos que possibilitam sua verificação, conforme ressalta Gil (2019). Além disso, compreender os métodos disponíveis é fundamental para conduzir a investigação de forma adequada, garantindo que os procedimentos adotados legitimem o saber adquirido e conduzam a uma pesquisa válida e confiável.

3.1 APRESENTAÇÃO DO CASO

A característica metodológica deste trabalho é um estudo documental, centrado na análise de dados já existentes, oriundos das atividades de um projeto de Pesquisa e Desenvolvimento na área de capacitação, em Manaus/Amazonas.

De acordo com Junior et al. (2021), a pesquisa documental distingue-se da pesquisa bibliográfica por utilizar principalmente fontes primárias, ou seja, documentos originais que ainda não foram analisados de forma sistemática. Essa abordagem permite ao pesquisador examinar diretamente os registros que possuem relação com o objeto de estudo.

Segundo Gil (2019, p. 62), a pesquisa documental guarda semelhanças com a bibliográfica, mas se diferencia por empregar materiais que ainda não passaram por um tratamento analítico, ou que podem ser reelaborados conforme os objetivos da investigação.

A Startup X, que motivou este estudo, apresenta características comuns às startups em geral. Fundada por duas sócias jovens, ambas em formação em engenharia elétrica, a empresa tem por objetivo atuar na área de serviços para o segmento de varejo. De acordo com registros do programa de desenvolvido no projeto de P&D examinado, a startup buscou adquirir maturidade em áreas essenciais para o crescimento do negócio, como gestão, marketing e desenvolvimento de produto, aprendendo conforme as demandas surgem, e opera com recursos limitados, tanto em termos de equipe — composta por seis colaboradores — quanto em capacidade de produção, especialmente no que diz respeito ao desenvolvimento de seu primeiro produto.

Durante sua participação no programa de desenvolvido no projeto de P&D examinado, a Startup X passou por um processo de evolução significativo. Registros indicam que o MVP inicial, embora funcional, foi considerado de baixa funcionalidade, evidenciando limitações técnicas e de usabilidade que poderiam comprometer sua aceitação no mercado. A partir desse diagnóstico, a equipe passou a investir na reestruturação do produto, incorporando melhorias com base no feedback recebido de mentores e potenciais usuários. Esse movimento foi essencial para alinhar o desenvolvimento do MVP às reais necessidades do público-alvo, tornando-o mais robusto e competitivo.

Além da evolução técnica, o programa de desenvolvido no projeto de P&D examinado proporcionou à startup uma imersão em áreas estratégicas como gestão empresarial, marketing digital e metodologias ágeis de desenvolvimento. As sócias e os demais membros da equipe passaram a aplicar ferramentas de planejamento e validação de mercado, o que contribuiu para uma visão mais clara do modelo de negócio e das etapas necessárias para sua escalabilidade. Com isso, a Startup X não apenas aprimorou seu produto, mas também fortaleceu sua estrutura organizacional, demonstrando maior preparo para enfrentar os desafios do ecossistema de inovação onde está inserida.

3.2 DESCRIÇÃO DO CONTEXTO ESTUDADO

No início do programa de pré-incubação, o engajamento da equipe da Startup X era considerado baixo, refletindo a inexperiência e os desafios enfrentados por um grupo ainda em formação. Com o passar do tempo, no entanto, esse cenário começou a mudar. A participação ativa nas atividades do programa, aliada ao amadurecimento das fundadoras e colaboradores, contribuiu para uma maior integração e comprometimento com os objetivos da startup, transformando-os em uma equipe mais coesa, motivada e alinhada com a proposta de inovação e crescimento sustentável do negócio.

A equipe é composta por duas sócias fundadoras e quatro colaboradores com funções bem definidas: um Scrum Master, responsável por organizar os fluxos de trabalho e promover a metodologia ágil; um CTO (Chief Technology Officer), encarregado do desenvolvimento técnico e da evolução do produto; um CFO (Chief Financial Officer), que cuida da gestão financeira e planejamento estratégico; e um CMO (Chief Marketing Officer), responsável pelas ações de comunicação e posicionamento da marca. Essa estrutura tem permitido à Startup X avançar de forma mais organizada e eficiente, com cada membro contribuindo dentro de sua área de especialidade para o fortalecimento do projeto.

O produto desenvolvido pela Startup X consiste em uma plataforma inovadora voltada ao setor de varejo alimentar, com o objetivo de conectar consumidores às grandes redes varejistas. A solução busca otimizar a experiência de compra por meio da integração de dados e da oferta de funcionalidades que facilitam o acesso a informações relevantes sobre produtos, promoções e comportamento de consumo. A proposta é promover uma relação mais eficiente entre clientes e empresas, utilizando tecnologia como meio de aproximação e fidelização.

Conforme registros do projeto de desenvolvido no projeto de P&D examinado, além da interface voltada ao consumidor, a plataforma visa oferecer recursos robustos de gerenciamento e análise de dados para os varejistas. Por meio de dashboards e relatórios personalizados, é possível acompanhar métricas de desempenho, comportamento de compra, preferências dos clientes e indicadores operacionais. Essa abordagem orientada por dados permite decisões mais estratégicas, contribuindo para a melhoria contínua dos serviços e para o aumento da competitividade no mercado.

3.3 APLICAÇÃO DOS CONCEITOS ÁGEIS NA CONSTRUÇÃO DA METODOLOGIA HÍBRIDA.

De acordo com Sutton (2000), o desenvolvimento de software ocorre por meio de processos, ainda que muitas vezes de forma improvisada. Por se tratar de uma atividade pertencente à engenharia de software, exige certo grau de rigor e organização. A criação e a repetição de processos contribuem significativamente para o amadurecimento das startups, refletindo diretamente em seu sucesso. Nesse contexto, a metodologia assume papel essencial, pois representa o conjunto de métodos, etapas e técnicas que orientam a pesquisa e o desenvolvimento, organizando e sistematizando a coleta e análise de dados, conforme destacam Lakatos e Marconi (2021).

Com o objetivo de aprimorar os processos da Startup X e alcançar melhores resultados no gerenciamento e desenvolvimento de software, foi proposta por mentores do programa de desenvolvido no projeto de P&D examinado uma metodologia híbrida, pela combinação dos princípios das abordagens em cascata e ágil. Essa abordagem, baseada em metodologias previamente apresentadas, foi desenhada para ser suficientemente simples para que a *startup* faça a implantação e valide se as metodologias geram ou não benefícios para seu modelo de negócio.

Escolher uma metodologia híbrida, em vez da adoção exclusiva de uma abordagem específica, deve-se às limitações e dificuldades enfrentadas por startups na aplicação de determinadas práticas, como a programação em pares, por exemplo. A proposta visa adaptar-se à realidade da Startup X, conciliando práticas eficazes de diferentes metodologias de forma flexível e funcional.

Na Tabela 3, são apresentados os conceitos e práticas das metodologias analisadas, juntamente com suas classificações na metodologia proposta — indicando se estão presentes, ausentes ou são opcionais. Em seguida, são apresentadas as justificativas para cada uma dessas classificações, com base na aplicabilidade e viabilidade dentro do contexto da startup.

QUADRO 3: CONCEITOS E PRÁTICAS DAS METODOLOGIAS PROPOSTAS

Prática	Origem	Objetivo Principal	Contribuição para o desenvolvimento do produto da startup
Cliente Presente	XP	Facilitar comunicação direta com o cliente	Adaptada para simular feedback via MVPs e registros no Trello
Código Coletivo	XP	Permitir que todos tenham acesso ao código	Favorece colaboração e agilidade na construção e manutenção do sistema
Padrão de Código	XP	Garantir consistência e legibilidade do código	Facilita integração contínua e onboarding de novos membros
Projetos Simples	XP	Reduzir complexidade e facilitar manutenção	Aumenta a eficiência e a compreensão do sistema
Sprint	Scrum	Ciclos curtos de entrega de valor	Promove entregas frequentes e controle de progresso
Reunião Diária	Scrum	Alinhar atividades diárias da equipe	Melhora a comunicação e identifica impedimentos rapidamente
Backlog do Produto	Scrum	Listar funcionalidades e requisitos do produto	Organiza o escopo e prioriza entregas
Backlog da Sprint	Scrum	Planejar tarefas do ciclo atual	Garante foco e previsibilidade nas entregas
MVP	Lean	Validar soluções com menor esforço e tempo	Permite feedback rápido e ajustes às necessidades do mercado
Integração Contínua	XP	Atualizar o sistema com pequenas mudanças frequentes	Reduz erros e aumenta a estabilidade do sistema
Planning Poker	XP	Estimar o esforço para a criação das funcionalidades	Facilita manutenção e evolução do sistema
Semana de 40 horas	XP	Evitar sobrecarga e manter produtividade sustentável	Preserva a saúde da equipe e melhora a qualidade das entregas

FONTES: O AUTOR (2025)

Considerando as características e o modelo de negócio da Startup X, a metodologia híbrida proposta exige a participação ativa e constante do cliente ao longo do processo de desenvolvimento. Isso se deve ao fato de que a startup já possui um público-alvo definido e passou pela fase de validação do modelo de negócio, que se mostrou sustentável, repetível e escalável. A presença do cliente é essencial para garantir que as soluções desenvolvidas estejam alinhadas às suas reais necessidades, permitindo ajustes rápidos e direcionados com base em feedbacks contínuos.

A propriedade coletiva do código é uma prática comum em startups iniciantes, principalmente devido ao número reduzido de integrantes nas equipes e à centralização dos projetos. Nessa abordagem, todos os membros possuem liberdade para modificar qualquer parte do código, assumindo a responsabilidade de garantir seu correto funcionamento após as e sua integração/integridade. A estrutura enxuta

das equipes favorece a comunicação e a colaboração, tornando essa prática natural e eficiente.

Assim como ocorre no XP, a propriedade coletiva do código está fortemente relacionada à padronização da codificação. Por esse motivo, ambas as práticas foram incorporadas à metodologia híbrida. A existência de um padrão de codificação facilita a compreensão do código por todos os membros da equipe, além de reduzir a curva de aprendizado para novos colaboradores. Essa prática deve ser adotada de forma voluntária, e seguida com rigor para garantir a consistência e a qualidade do código, tanto em sua criação quanto em sua manutenção.

Outra prática selecionada para compor a metodologia híbrida é o projeto simples. Essa abordagem é essencial, pois contribui para a eficácia e para a manutenção do propósito. A simplicidade de planejamento deve ser buscada em todas as etapas do projeto, sem comprometer a qualidade. Isso significa encontrar soluções eficientes para os problemas e implementar novas funcionalidades de forma que a estrutura do plano permaneça clara, adaptável e de fácil manutenção.

Os sprints possuem uma duração previamente definida, assim como as tarefas que devem ser executadas dentro desse período. Com base na reunião de planejamento, para ao final do ciclo, entregar uma versão funcional do produto.

A reunião diária, é uma prática oriunda do Scrum e foi selecionada para compor a metodologia híbrida. Seu principal objetivo é revisar o progresso desde a última reunião anterior, alinhar as atividades da equipe e planejar as ações para as próximas 24 horas. Além disso, essa reunião permite que os líderes acompanhem de perto o andamento das tarefas, identifiquem possíveis obstáculos e direcionem esforços para os pontos que exigem maior atenção. Com duração de apenas alguns minutos por dia, a reunião diária oferece uma visão clara sobre o que cada membro da equipe está executando. Essa prática facilita o monitoramento do progresso, a identificação de gargalos no processo de desenvolvimento e o ajuste rápido de prioridades. Além disso, contribui para o controle eficaz dos requisitos e tarefas em andamento, ajudando a garantir que as estimativas feitas anteriormente sejam cumpridas dentro dos prazos estabelecidos.

O backlog do produto é um artefato importante para que a equipe tenha conhecimento das funcionalidades e tarefas que serão necessárias desenvolver e incorporar ao produto, tornando-o utilizável pelos usuários. Como em startups o produto pode não ser conhecido no início do projeto, os itens do backlog do produto

podem sofrer alterações à medida que o produto vai evoluindo. Tarefas e funcionalidades podem ser removidas ou adicionadas de acordo com a opinião dos usuários que testaram a aplicação com as funcionalidades já existentes. O mesmo é válido para o backlog do produto.

O backlog da sprint auxilia a equipe a nortear e saber quais tarefas irá realizar no próximo ciclo de trabalho. É caracterizado por um conjunto de itens do backlog do produto, juntamente com o plano para criar um incremento "Pronto" e atingir a meta da sprint. Ele é criado durante a reunião de planejamento da sprint e funciona como um plano de trabalho em tempo real, que ajuda a equipe a se manter focada e a priorizar a atividade. A equipe de desenvolvimento é responsável por criar, manter e atualizar o backlog da sprint ao longo do trabalho.

A técnica do MVP foi incorporada à metodologia por se tratar de uma abordagem eficaz para que startups validem se a solução que pretendem lançar no mercado realmente atende às expectativas e necessidades de seu público-alvo. Ao compreenderem o conceito de MVP e desenvolverem uma versão inicial e simplificada do produto, as startups conseguem apresentar essa versão aos usuários e coletar feedback valioso. Esse retorno permite direcionar os esforços de desenvolvimento para as funcionalidades que realmente agregam valor, evitando o desperdício de tempo e recursos com requisitos que não serão utilizados ou que não despertam interesse nos usuários. Dessa forma, a aplicação do MVP está alinhada aos princípios do Lean Startup, promovendo um desenvolvimento mais ágil, eficiente e centrado no usuário.

A integração contínua também é uma prática fundamental no dia a dia das equipes de desenvolvimento, principalmente quando se trabalha com tarefas pequenas e incrementais. Ao finalizar cada tarefa — incluindo desenvolvimento e testes —, o código deve ser imediatamente integrado ao repositório principal. Essa abordagem evita o acúmulo de mudanças não testadas e reduz o risco de erros se propagarem pelo sistema, facilitando a identificação e correção de falhas em estágios iniciais do projeto.

A entrega de versões utilizáveis do sistema em intervalos curtos é essencial, especialmente no contexto de startups. Essas empresas, ao buscarem soluções inovadoras e ainda inexistentes no mercado, precisam validar rapidamente se as funcionalidades desenvolvidas atendem às necessidades e expectativas dos potenciais usuários e stakeholders.

Com entregas frequentes aliadas ao projeto simples, alterações de requisitos não causam impactos muito grandes ou prejudiciais ao projeto como um todo. Além disso, ajudam a identificar as funcionalidades importantes do sistema e a definir a prioridade de desenvolvimento das funcionalidades presentes no backlog do produto para as próximas iterações do projeto.

Na reunião de planejamento da sprint, os membros da equipe recebem as informações que precisam sobre os requisitos que deverão desenvolver na sprint atual. Para que a equipe possa estimar o tempo e esforço que será empregado podem fazer uso do Planning Poker, quando todos os envolvidos nas histórias de desenvolvimento participam, indicando sua visão a respeito das histórias, levando em consideração o esforço e o tempo que serão necessários para desenvolver tal item. Após a avaliação individual, cabe a cada membro explicar o motivo de sua escolha, para que cheguem a um denominador em comum (Ritter, 2014).

No Planning Poker, cada integrante da equipe recebe um conjunto de 12 cartas numeradas de forma semelhante à sequência de Fibonacci, como ilustrado na Figura 4. Essas cartas geralmente apresentam os seguintes valores: 0, 1, 2, 3, 5, 8, 13, 20, 40, 100, além de uma carta com o símbolo de interrogação (?), utilizada quando o participante não se sente apto a estimar a tarefa em questão. Durante a dinâmica, cada tarefa prevista para a sprint é apresentada à equipe, e todos os participantes escolhem, de forma simultânea e sigilosa, a carta que representa sua estimativa de esforço ou complexidade para a execução da atividade. Após a revelação das cartas, discutem-se as diferenças entre as estimativas, buscando compreender os motivos por trás das variações. O objetivo é alcançar um consenso sobre o valor mais adequado para representar o esforço necessário. Quando uma tarefa recebe uma pontuação superior a 20, isso geralmente indica que se trata de uma história extensa e complexa, chamada de épico, sendo recomendável dividi-la em histórias menores e mais específicas, a fim de facilitar sua compreensão, execução e acompanhamento, conforme orienta Ritter (2014).

Figura 4: Cartas de Planning Poker

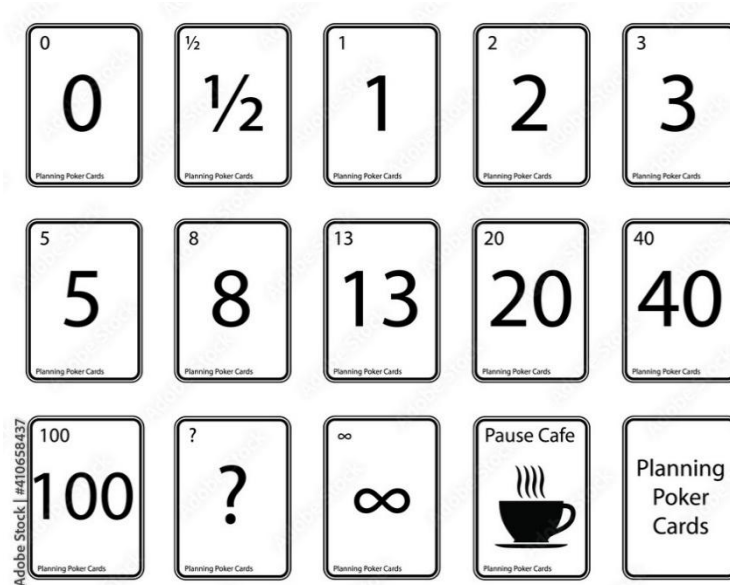


FIGURA 3: CARTAS DE PLANNING POKER (FONTE: [HTTP://STOCK.ADOBE.COM/BR/IMAGES/PLANNING-POKER-CARDS/410658437](http://stock.adobe.com/br/images/planning-poker-cards/410658437)) (2025)

A prática da semana de 40 horas, originária do XP, foi incorporada à metodologia proposta. Segundo Teles (2004), a fadiga dos colaboradores pode gerar efeitos negativos, como a queda na qualidade do trabalho e o uso ineficiente do tempo durante o expediente regular. A adoção dessa prática contribui para evitar o desgaste da equipe, ao desencorajar jornadas prolongadas e incentivar o descanso adequado, incluindo folgas regulares. Essa abordagem também está alinhada aos princípios do Lean Startup, ao evitar desperdícios no processo de desenvolvimento.

3.4 ANÁLISE DA APLICAÇÃO DA METODOLOGIA HÍBRIDA

Após a definição das práticas que compõem a metodologia híbrida, o passo seguinte foi sua implantação por parte da Startup X, com o objetivo de verificar se o conjunto de práticas selecionado traria melhorias reais aos processos de desenvolvimento e gestão da empresa. A proposta visava alinhar práticas ágeis e enxutas à realidade da startup, promovendo maior organização, controle e eficiência no desenvolvimento de software. Entretanto, devido limitações de tempo e à necessidade da empresa em concluir acordos comerciais e projetos em andamento, a implantação da metodologia não pôde ser realizada em sua totalidade. Como resultado, a análise e validação da metodologia não foram completas, impossibilitando uma conclusão definitiva sobre sua eficácia. Ainda assim, a adoção parcial de

algumas práticas permitiu observar indícios de melhorias nos processos internos da empresa.

Entre as práticas efetivamente adotadas, destacam-se as sprints, as reuniões diárias, a integração contínua, o uso do MVP e a organização do backlog da sprint.

Apesar da ausência de métricas anteriores que possibilitassem uma comparação quantitativa entre os resultados anteriores e os atuais, foi observada evolução em diversos aspectos como demonstra na Tabela 4.

QUADRO 4: TABELA ANALÍTICA DE RESULTADOS

Prática	Objetivo Principal	Objetivo Alcançado
Cliente Presente	Facilitar comunicação direta com o cliente	A equipe estabeleceu canais de comunicação com usuários reais, permitindo ajustes rápidos no MVP com base em feedbacks coletados semanalmente.
Código Coletivo	Permitir que todos acessem e editem o código	Implementação de repositório compartilhado no GitHub, com controle de versões e revisões colaborativas entre os desenvolvedores.
Padrão de Código	Garantir consistência e legibilidade do código	Foi realizada a criação de um guia interno de estilo de codificação, utilizado por todos os membros da equipe técnica.
Projetos Simples	Reduzir complexidade e facilitar manutenção	A divisão do sistema em módulos independentes, facilitou testes e atualizações organizando o funcionamento geral.
Sprint	Ciclos curtos de entrega de valor	Realização de sprints quinzenais auxiliaram as entregas funcionais e revisão de metas atingidas.
Reunião Diária	Alinhar atividades diárias da equipe	Foram realizadas reuniões diárias de 15 minutos para alinhamento de tarefas e andamento de projeto.
Backlog do Produto	Listar funcionalidades e requisitos do produto	Organização do backlog no Trello auxiliou na organização e registros de valor de negócio e feedback dos usuários.
Backlog da sprint	Planejar tarefas do ciclo atual	O Planejamento semanal das tarefas foi realizado com base nas metas da sprint e na capacidade da equipe.
MVP	Validar soluções com menor esforço e tempo	Realizado lançamento de versão inicial funcional da plataforma, utilizada em testes com usuários reais para validação de funcionalidades.
Integração Contínua	Atualizar o sistema com pequenas mudanças frequentes	O processo garante que o sistema da startup seja atualizado com frequência,

		com menos erros, mais segurança e mais rapidez.
Planning Poker	Estimar esforço das tarefas de forma colaborativa	Utilização de Planning Poker para estimar tarefas durante as reuniões de planejamento, promoveu consenso entre os membros.
Semana de 40 horas	Evitar sobrecarga e manter produtividade sustentável	Houve menos retrabalho e maior foco nas tarefas.

FORNE: O AUTOR (2025)

A introdução das sprints e das reuniões diárias contribuiu significativamente para o aprimoramento do gerenciamento das atividades da equipe. Os sócios da empresa passaram a ter maior visibilidade sobre o andamento dos trabalhos, o que lhes permitiu dedicar mais tempo a outras áreas estratégicas do negócio, com a confiança de que as entregas planejadas seriam concluídas ao final de cada sprint em versões utilizáveis do sistema.

A técnica do MVP também foi incorporada com sucesso, embora a entrega de uma versão mínima viável nem sempre ocorra ao final de uma única sprint. Em alguns casos, são necessários múltiplos ciclos para que uma versão com valor real para o usuário seja concluída. Após atingir esse ponto, a startup apresentou o MVP a um usuário-chave, que validou a proposta. Um teste foi realizado, os resultados foram fundamentais para o refinamento dos requisitos. Esse processo permitiu identificar funcionalidades mais relevantes, eliminar aquelas com baixa aceitação e ajustar o backlog do produto e da sprint com base no feedback real dos usuários.

Para armazenar os itens do backlog do produto e da sprint, a equipe começou utilizando a ferramenta Trello, onde é possível simular um quadro Kanban, adicionando colunas e cartões com as atividades a serem desenvolvidas, como pode ser visto na Figura 5. Nos cartões ainda é possível definir subtarefas e adicionar comentários para a tarefa, o que facilita a descrição do que deve ser realizado na tarefa e descreve o que foi feito para que a tarefa chegasse ao estado de “pronto”. Outra funcionalidade da ferramenta é a possibilidade de utilizar etiquetas com diferentes cores, que podem ser usadas para direcionar tarefas a determinado membro da equipe. No Trello também é possível adicionar a cada tarefa sua pontuação recebida no planning poker, e assim ter uma estimativa, por coluna, dos pontos recebidos pelas tarefas e qual o status da pontuação (quantos pontos por coluna).

FIGURA 5: EXEMPLO DE KANBAN VIA TRELLO



FONTE: TRELLO.COM (2025)

Com a adoção da ferramenta Trello, a equipe de desenvolvimento passou a contar com uma estrutura visual que funciona como painel de documentação dinâmica dos requisitos. Essa organização facilita a compreensão das tarefas a serem executadas, reduzindo significativamente o risco de esquecimentos e a perda de tempo na busca por informações complementares. A clareza na descrição das atividades e a possibilidade de acompanhar o progresso de cada tarefa contribuem diretamente para a eficiência do processo de desenvolvimento.

Foi observado que o Trello também se mostrou uma ferramenta estratégica para o acompanhamento da evolução do projeto. A visualização dos pontos atribuídos a cada tarefa, por meio do Planning Poker, e o somatório por coluna permitem uma análise rápida do andamento dos sprints. Com esses dados, é possível identificar gargalos, redistribuir esforços e definir planos de ação mais assertivos, evitando atrasos e promovendo maior previsibilidade nas entregas.

Mesmo com a implementação, ainda parcial das práticas propostas pela metodologia híbrida, já foi possível observar melhorias significativas nos processos de desenvolvimento da Startup X. A estruturação das sprints, aliada às reuniões diárias e à integração contínua, trouxe maior controle sobre as atividades e facilitou o gerenciamento do tempo e dos recursos. Com a repetição dessas práticas e a futura incorporação das demais etapas previstas na metodologia, a empresa poderá estabelecer indicadores de desempenho mais precisos, permitindo mensurar a

velocidade de desenvolvimento da equipe e estimar com maior confiabilidade o tempo necessário para implementar novos requisitos no sistema. Esses avanços são fundamentais para a maturação dos processos internos e para a construção de uma base sólida que sustente o crescimento da startup.

4 RESULTADOS E DISCUSSÃO

A aplicação parcial da metodologia híbrida na Startup X permitiu observar melhorias concretas nos processos de desenvolvimento e gestão. A adoção de práticas específicas, mesmo que de forma gradual, trouxe ganhos em organização, comunicação e entrega de valor.

A estruturação do trabalho em sprints quinzenais, aliada às reuniões diárias, proporcionou maior controle sobre o andamento das tarefas e facilitou o alinhamento entre os membros da equipe. Essa dinâmica contribuiu para a previsibilidade das entregas e para a identificação ágil de impedimentos.

O uso do Trello como ferramenta de gestão visual, com quadros no estilo Kanban, permitiu organizar o backlog do produto e da sprint de forma clara e acessível. A categorização das tarefas, o uso de etiquetas e a visualização dos story points atribuídos via Planning Poker facilitaram a priorização e o acompanhamento do progresso, promovendo maior eficiência na execução das atividades.

A prática da integração contínua foi incorporada com sucesso, garantindo atualizações frequentes e seguras no sistema. Essa abordagem reduziu falhas e aumentou a estabilidade do produto, mesmo com uma equipe enxuta.

A entrega de MVPs funcionais possibilitou a coleta de feedbacks reais de usuários, que foram incorporados ao backlog, permitindo ajustes rápidos e direcionados. Essa estratégia reforçou o foco no valor entregue ao cliente e na adaptação contínua do produto às necessidades do mercado.

Por fim, a adoção da jornada de trabalho sustentável (sem extrapolação de horas) resultou em maior foco e menor retrabalho, favorecendo a qualidade das entregas e o bem-estar da equipe.

Esses resultados, ainda que iniciais, indicam que a metodologia híbrida proposta é compatível com a realidade da startup e tem potencial para impulsionar sua maturidade organizacional. A continuidade da aplicação das práticas e a definição de métricas de desempenho serão fundamentais para consolidar os avanços observados.

5 CONCLUSÃO

A presente pesquisa demonstrou que a aplicação de uma metodologia híbrida, baseada em práticas ágeis e enxutas, pode contribuir significativamente para a organização e o desempenho de startups em estágio inicial. A experiência da Startup X evidenciou que, mesmo equipes sem experiência na implementação de metodologias ágeis podem observar avanços concretos na gestão de projetos, na comunicação da equipe e na entrega de valor ao cliente.

A estruturação do trabalho em ciclos curtos, o uso de ferramentas visuais de acompanhamento e a validação contínua por meio de MVPs permitiram maior previsibilidade, foco e alinhamento com as necessidades do mercado. A integração contínua e a padronização do código reforçaram a qualidade técnica do produto, enquanto a jornada de trabalho sustentável favoreceu o bem-estar da equipe e a consistência das entregas.

Embora a metodologia não tenha sido aplicada em sua totalidade, os resultados obtidos indicam que a proposta é promissora e tem potencial de aplicação em startups em estágio inicial e com recursos limitados. A continuidade da aplicação das práticas e o monitoramento sistemático de indicadores de desempenho poderão consolidar os ganhos observados, contribuindo para a maturação dos processos internos e para o fortalecimento da posição da startup no mercado.

Dessa forma, conclui-se que a combinação de metodologias ágeis, quando adaptada ao contexto específico de uma startup, representa uma estratégia eficaz para enfrentar os desafios de gestão e desenvolvimento de projetos em ambientes de elevada incerteza.

REFERÊNCIAS

AGHINA, Wouter et al. Agile Project Delivery Confidence. PwC, 2017.

AGUIAR, Felipe; CAROLI, Paulo. Product Backlog Building (PBB). 2024.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE. Disponível em: <https://mercado.abes.org.br/>. Acesso em: 04 jul 2025.

BECK, Kent et al. Manifesto Ágil. 2001.

BECK, Kent. Extreme Programming Explained: Embrace Change. 2. ed. Boston: Addison-Wesley, 2004.

CARVALHO, João; ABRANTES, José; CAMEIRA, Manuel. Metodologias Ágeis: Uma Abordagem Prática. Lisboa: FCA, 2011.

DIAS, Cláudio. Metodologias Ágeis: Uma Abordagem Prática. São Paulo: Novatec, 2010.

FIGLIARELLI, Vinícius. In: GOMES, Antônio; ARAÚJO, João. Metodologia Científica. São Paulo: Atlas, 2005.

FOWLER, Martin. The New Methodologies. 2005.

GIL, Antonio Carlos. Como Elaborar Projetos de Pesquisa. 6. ed. São Paulo: Atlas, 2019.

GITAHY, Bruno. Startup: Manual do Empreendedor. São Paulo: Editora do Autor, 2010.

GITAHY, Bruno. Startup: Manual do Empreendedor. São Paulo: Editora do Autor, 2015.

HILMAN, John. Agile Software Development: A Comprehensive Guide. 2004.

JUNIOR, Carlos et al. Pesquisa Documental: Fundamentos e Aplicações. 2021.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. Fundamentos de Metodologia Científica. 8. ed. São Paulo: Atlas, 2021.

LEITÃO, Fernando. Gestão Ágil de Projetos. 2010.

LUDVIG, Marcos; REINERT, Marcelo. Metodologias Ágeis: Uma Nova Abordagem. 2007.

MAHINIC, Marko; HOVELJA, Tomaž. Agile Estimation Techniques. 2012.

MALLIDI, Sandeep; SHARMA, Ankit. Planning Poker and Agile Estimations. 2021.

MUNIALO, S. W.; MUKETHA, G. M. A review of agile software effort estimation methods. International Journal of Computer Applications Technology and Research, v. 5

OHNO, Taiichi. Toyota Production System: Beyond Large-Scale Production. Portland: Productivity Press, 1997.

PANTALIÃO, Rodrigo. O que é Lean?. 2009.

POPPENDIECK, Mary; POPPENDIECK, Tom. Lean Software Development: An Agile Toolkit. Boston: Addison-Wesley, 2003.

RITTER, Fernando. Planning Poker: Estimativas Ágeis. 2014.

SATO, Marcelo. Metodologias Ágeis no Desenvolvimento de Software. 2007.

SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide. 2011.

SHINGO, Shigeo. A Study of the Toyota Production System. 1996.

SOMMERVILLE, Ian. Software Engineering. 9. ed. Boston: Pearson, 2011.

SUTHERLAND, Jeff et al. The Scrum Papers: Nuts, Bolts, and Origins of an Agile Framework. 2000.

SUTTON, Robert. Software Process Improvement in Startups. 2000.

The impact of agility: How to shape your organization to compete.

TAKEUCHI, Hirotaka; NONAKA Ikujiro. Disponível em: <https://hbr.org/1986/01/the-new-new-product-development-game>. Acesso em 10 set 2025.

TELES, Vinícius. Extreme Programming: Práticas para o Desenvolvimento Ágil. 2004.

VIEIRA, Marcelo. Scrum: Guia Prático. 2014.