

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA
PROJETO DE CONCLUSÃO DE CURSO

ANDRESSA SANTARÉM FERREIRA

**APLICAÇÃO DE METODOLOGIAS ÁGEIS DE GESTÃO NO
DESENVOLVIMENTO DE PROJETOS DE TECNOLOGIA PARA
ACELERAÇÃO DE ENTREGAS, VALIDAÇÃO E MELHORA NA
INTEGRAÇÃO DA EQUIPE**

Uberlândia – MG

2022

ANDRESSA SANTARÉM FERREIRA

**APLICAÇÃO DE METODOLOGIAS ÁGEIS DE GESTÃO NO
DESENVOLVIMENTO DE PROJETOS DE TECNOLOGIA PARA
ACELERAÇÃO DE ENTREGAS, VALIDAÇÃO E MELHORA NA
INTEGRAÇÃO DA EQUIPE**

Trabalho de Conclusão de Curso
apresentado como requisito da disciplina de
Projeto de Fim de Curso II, do 10º período
da graduação em Engenharia Mecatrônica,
da Faculdade de Engenharia Mecânica,
Universidade Federal de Uberlândia – UFU,
Campus Glória
Orientador: Prof. Doutor Eustáquio São
José de Faria

Uberlândia – MG

2022

AGRADECIMENTO

Quero agradecer primeiro e principalmente aos meus pais, Adriana e Cláudio e à minha irmã Cláudia. Obrigada por sempre acreditarem em mim, me apoiarem e, principalmente, pela paciência. Tudo que eu sou é graças a vocês.

Ao Cairo, por sempre estar aqui.

À empresa X por me oferecer o suporte e os dados necessários para a conclusão desse projeto.

Ao Mário, por me ajudar a colocar as ideias em ordem.

Ao Prof. Dr. Eustáquio São José de Faria, pela orientação e pelos ensinamentos.

E aos amigos e familiares que trilharam comigo esse caminho.

RESUMO

Este projeto teve como objetivo estudar as formas como são aplicadas metodologias ágeis e metodologias tradicionais de forma híbrida com o intuito de gerenciar os projetos de engenharia de software de uma empresa de TI. Foi estudada a utilização do método *Scrum*, com todos seus rituais de reuniões (planejamento, diárias, de fechamento e retrospectiva) adaptados para a realidade do mundo de TI, com o apoio de ferramentas como Kanban e o gráfico de *Burndown* para melhor visualização e acompanhamento das entregas durante as *sprints*. A utilização do Método Cascata, em um primeiro momento, se fez necessária, para entregar um valor de projeto para os *stakeholders* além, também, de entregar uma noção de duração deste. A utilização híbrida das metodologias de gestão de projetos se mostra útil quando aparecem pendências ou alterações de escopo do projeto. Nesse contexto, a identificação de problemas se dá de maneira mais rápida e efetiva e as mudanças necessárias são feitas de forma ágil. O projeto também visa expor possíveis dificuldades enfrentadas pela equipe *Scrum* durante o seu desenvolvimento.

Palavras-chave: Metodologia ágil, *Scrum*, método cascata, *Kanban*, gráfico de *burndown*, *sprints*.

ABSTRACT

This project aimed to study the ways in which agile methodologies and traditional methodologies are applied in a hybrid way in order to manage the software engineering projects of an IT company. The use of the Scrum method with all its meeting rituals (planning, dailys, sprint closing and sprint retrospective) adapted to the reality of the IT world, with the support of tools such as Kanban and the Burndown chart for better visualization and monitoring of deliveries during the sprints and the use of the Waterfall Method, at first to deliver a project value to the stakeholders in addition to delivering a notion of project duration. The hybrid use of project management methodologies proves to be useful when pending issues or changes in the project scope appear, the identification of problems takes place more quickly and effectively and the necessary changes are made in an agile way. The project also aims to expose possible difficulties faced by the Scrum team during the development of the project.

Keywords: agile methodologies, Scrum, Waterfall Method, Kanban, burndown chart, sprints.

LISTA DE FIGURAS

Figura 1 - Principais artefatos do Scrum	19
Figura 2 - Fases da metodologia Cascata.....	24
Figura 3 - Exemplo de Quadro Kanban.....	26
Figura 4 - Exemplo de Gráfico de <i>Burndown</i>	27
Figura 5 - Processo de contratação de novos projetos.	30
Figura 6 - Interface do aplicativo de criação de cronogramas	32
Figura 7 - Cronograma Inicial do Projeto Z	32
Figura 8 - Mapa Mental da Quebra de atividades para o <i>Backlog</i> do Produto .	35
Figura 9 - <i>Backlog</i> do Produto Priorizado.....	36
Figura 10 - Gráfico de <i>Burndown</i> da primeira <i>Sprint</i>	39
Figura 11 - Quadro Kanban da Primeira Sprint	40
Figura 12 - Cronograma da Mudança de Escopo.....	43
Figura 13 - Atividades entregues ao longo das sprints.....	45

LISTA DE QUADROS

Quadro 1 - Estrutura da equipe	33
Quadro 2 - Horas por <i>sprint</i> da equipe	36
Quadro 3 – Divisão das atividades da primeira <i>sprint</i> para equipe	37
Quadro 4 - Configuração da Equipe após adição de recurso.....	42

LISTA DE ABREVIATURAS E SIGLAS

- PO – ProductOwner
- SM – Scrum Master
- TI – Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	14
1.1.1	<i>Objetivo Geral.....</i>	<i>14</i>
1.1.2	<i>Objetivos Específicos</i>	<i>14</i>
1.2	JUSTIFICATIVA.....	15
1.3	ESTRUTURA DO TRABALHO	15
2	REVISÃO BIBLIOGRÁFICA	17
2.1	GERENCIAMENTO DE PROJETOS.....	17
2.2	METODOLOGIAS LEVES	18
2.2.1	<i>Scrum</i>	<i>18</i>
2.2.1.1	A metodologia.....	20
2.2.1.2	A Equipe	20
2.2.1.2.1	Desenvolvedores	20
2.2.1.2.2	Scrum Master.....	20
2.2.1.2.3	Product Owner	21
2.2.1.3	<i>Sprints</i>	<i>21</i>
2.2.1.4	Reuniões	22
2.2.1.4.1	Planejamento da Sprint.....	22
2.2.1.4.2	Reuniões diárias (dailys).....	22
2.2.1.4.3	Reunião de fechamento da sprint.....	23
2.2.1.4.4	Retrospectiva da sprint	23
2.3	METODOLOGIAS TRADICIONAIS	23
2.3.1	<i>Método Cascata.....</i>	<i>24</i>
2.4	KANBAN.....	25
2.5	GRÁFICO DE BURNDOWN	26
3	METODOLOGIA.....	28
4	DESENVOLVIMENTO DO PROJETO (ESTUDO DE CASO).....	29
4.1	ANÁLISE INICIAL	29
4.2	CRONOGRAMA	31
4.3	MONTANDO A EQUIPE.....	33
4.4	INÍCIO DO DESENVOLVIMENTO.....	34
4.4.1	<i>Quebra de atividades para Backlog do Produto.....</i>	<i>34</i>

4.4.2	<i>SprintPlanning</i>	37
4.4.3	<i>Reuniões Diárias (Dailys) da Primeira Sprint</i>	38
4.4.4	<i>Fechamento da Sprint e Retrospectiva</i>	40
4.5	<i>SPRINTS SEGUINTEs</i>	41
4.6	<i>DIFICULDADES ENFRENTADAS</i>	41
4.6.1	<i>Estimativas incorretas</i>	41
4.6.2	<i>Alteração de escopo</i>	43
4.6.3	<i>Reorganização da equipe</i>	44
4.6.4	<i>Atrasos</i>	44
4.6.5	<i>Entrega do Projeto</i>	45
5	CONCLUSÕES	47
6	TRABALHOS FUTUROS	48
7	REFERÊNCIAS BIBLIOGRÁFICAS	49
	FICHA DE AVALIAÇÃO	51

1 INTRODUÇÃO

Segundo Kuhrman *et al* (2021), existem diversas metodologias de gestão de projetos de software, as mais antigas datando de 1960. Com o passar dos anos, a demanda crescente por projetos de software exigiu a evolução das equipes para entregar os projetos com cada vez mais qualidade e rapidez. Com isso, métodos tradicionais como o método cascata e o espiral abriram espaço para as metodologias ágeis.

Nesse cenário constante de mudanças, para se tornarem competitivas, as empresas abraçaram cada vez mais a utilização de metodologias ágeis. Ainda assim, segundo Kuhrman *et al* (2021), metodologias puramente ágeis ainda são raras, apesar de estudos mostrarem que organizações híbridas em média possuem um percentual menor de agilidade. O que ocorre é a aplicação híbrida do ágil com o tradicional, por exemplo, entre o método cascata e o Scrum, misturando os aspectos que mais se adequem ao funcionamento da equipe e à visão da empresa.

De acordo com Ahmad, Soomro e Naqvi (2016), os métodos tradicionais têm seu foco no planejamento e definição do projeto inicialmente, com extensa documentação. Desde o início do projeto, é necessário ter requisitos e funcionalidades bem listadas e descritas, deixando pouca ou, em alguns casos, nenhuma decisão para a equipe desenvolvedora.

Enquanto isso, as metodologias ágeis têm seu foco na comunicação da equipe e em abraçar as mudanças necessárias durante o desenvolvimento – Nelas, o projeto é incrementado pelas entregas realizadas em períodos de tempo pré-determinados e decisões são tomadas a todo instante para atingir a totalidade do desenvolvimento.

Kuhrman *et al* (2021) pontuam que a diminuição da quantidade de documentação necessária para o desenvolvimento de softwares foi observada a partir da evolução das metodologias sequenciais até chegar ao manifesto ágil em 2001.

De acordo com Bishop, Rowland e Noteboom (2018) a preferência dos gestores de projetos por metodologias ágeis, entre outros motivos, se dá porque, na abordagem ágil, o gestor sai da posição de comando e controle para uma posição de facilitador, de modo a auxiliar o time a se comunicar melhor sem interferir no processo criativo. Forte e Kloppenborg (2018) pontuam também que, em um projeto ágil, o GP assume a postura de reagir aos problemas que aparecem durante o desenvolvimento, ao invés de ter tudo programado desde o início.

Segundo Diel *et al* (2015), no Brasil, o aumento na produtividade, a capacidade de adaptação diante de mudanças (repriorização de atividades) e a melhora na qualidade do software estão entre as maiores razões para a adoção de um método ágil. Nesse trabalho será analisado o desempenho de uma equipe na realização de um projeto na empresa de TI X, utilizando a combinação entre a metodologia *Scrum* e *Kanban*, em conjunto com o Método Cascata. Serão apresentados os problemas enfrentados durante o desenvolvimento e os resultados alcançados.

A Empresa X, de porte médio, atua em diversas áreas da TI, dentre elas, como uma fábrica de softwares no desenvolvimento de aplicações para ampla gama de clientes e áreas, tanto em formato web como *mobile*. Com a crescente demanda do mercado de softwares, a empresa X busca melhorar a performance das equipes desenvolvedoras ajustando os processos de gestão e desenvolvimento à metodologias ágeis.

Segundo Diel *et al* (2015), a falta de experiência com métodos ágeis, diferença entre a cultura da empresa e os valores do ágil, pressões externas para seguir metodologias tradicionais, falta de escopo e clientes treinados, além de baixas expectativas de gerenciamento são dificuldades enfrentadas para a aplicação do ágil em empresas brasileiras.

Por esses motivos, a aplicação do ágil pode ser um processo demorado e que exige esforço de diversas áreas da organização. Na empresa X, são utilizados métodos híbridos como forma de tentar vencer alguns desses impedimentos.

De acordo com Conboy e Fitzgerald (2004), tanto na academia quanto na indústria, ainda não existe um consenso sobre o que realmente é ágil nos métodos. Os autores consideram ainda que, para que um desenvolvimento seja realmente ágil, é necessário realizar muito mais do que atualmente é considerado.

Segundo Ahmad *et al* (2016), projetos ágeis são orientados à pessoas, adaptáveis, balanceiam flexibilidade e planejamento, possuem uma abordagem descentralizada, são simples e prezam pela colaboração.

Já Kuhrmann *et al* (2021) determinam que a agilidade é um conceito subjetivo que depende de quem observa o desenrolar do projeto. No artigo é realizada uma pesquisa para associar a forma como algumas práticas e métodos podem afetar o nível de agilidade do projeto. Mais da metade das técnicas analisadas foram direcionadas ao ágil e tentam se aproximar mais desse método, mesmo ainda utilizando métodos híbridos. O estudo ainda compara a forma como o manifesto ágil é filosófico e nem sempre mostra clareza enquanto os projetos desenvolvidos são uma parte concreta do dia a dia.

Bishop *et al* (2018) descrevem que as principais características positivas das metodologias ágeis são:

- Pragmatismo – faz com que o ágil se pareça mais com a natureza do trabalho em si, quando se trata do desenvolvimento de softwares;
- Leva em consideração a satisfação do cliente de forma a envolvê-lo no desenvolvimento, criando uma relação colaborativa;
- Cria uma melhor gerencia de riscos;
- Melhora a comunicação;
- Aumenta a satisfação do time;
- É feito de forma incremental – pois são realizadas pequenas entregas de tempos em tempos e
- É adaptativo.

Entre as características negativas, é observado o desejo por datas e estimativas bem definidas e a aversão às mudanças por parte do cliente. O estudo coloca como ponto neutro a possibilidade de adaptar e definir qual metodologia é mais adequada para cada tipo de projeto.

1.1 OBJETIVOS

Este trabalho se propõe a documentar as etapas de um projeto de engenharia de software e apresentar as ferramentas utilizadas para melhora da produtividade da equipe. Foi realizado o acompanhamento das entregas realizadas utilizando um híbrido entre os métodos de gestão *Scrum*, *Kanban* e *Cascata*.

1.1.1 Objetivo Geral

O objetivo geral é apresentar e realizar uma análise bibliográfica e documentação das etapas utilizadas por empresas de TI (Tecnologia da Informação) para entregar valor para seus clientes no desenvolvimento de projetos de engenharia de *software*, utilizando um híbrido de metodologias ágeis e tradicionais de forma adaptada, através da documentação das etapas de um projeto realizado na Empresa X.

1.1.2 Objetivos Específicos

Os objetivos específicos do projeto se compõem pela apresentação tanto das metodologias ágeis e das metodologias tradicionais, bem como uma breve introdução da história do gerenciamento de projeto. Serão apresentadas

também ferramentas que podem ser utilizadas como apoio na aplicação de tais metodologias.

Será apresentado o Projeto Z, realizado na Empresa X, sua estrutura de equipe e de organização de entregas de atividades para os *stakeholders*. Espera-se expor as dificuldades apresentadas durante o desenvolvimento e a forma como as metodologias auxiliam na melhor resolução dos problemas encontrados, de forma a comprometer o mínimo possível a entrega do projeto e também possibilitar mudanças de forma ágil assim que surge a necessidade.

1.2 JUSTIFICATIVA

O gerenciamento de projetos de software em empresas de TI é importante para garantir a entrega dos projetos com a qualidade e eficiência esperadas. Este trabalho visa mostrar, através do processo de pesquisa, a forma como o gerenciamento de projetos é moldável para atender às necessidades da equipe e empresa no desenvolvimento de projetos de *software*. A utilização de metodologias ágeis, em conjunto com metodologias tradicionais, é uma ferramenta que auxilia gerentes de projeto a adaptar seu gerenciamento para acompanhar a crescente evolução que as empresas de TI (Tecnologia da Informação) enfrentam atualmente. A aplicação de rituais da metodologia *Scrum* se mostram presentes para ajudar na resolução de problemas e em manter o tempo das entregas o mais otimizado possível, além de auxiliar na comunicação do time de desenvolvimento com o SM (*Scrum Master*), PO (*Product Owner*) e *stakeholders*.

1.3 ESTRUTURA DO TRABALHO

Para atingir os objetivos declarados anteriormente, é apresentada a revisão bibliográfica de artigos sobre o mesmo tema, em seguida a

documentação da observação do desenvolvimento do projeto (estudo de caso), onde é mostrado todo o processo para melhoria da produtividade da equipe. É mostrado como o método híbrido é utilizado no cenário de uma empresa de software e depois expostas as dificuldades enfrentadas. Em seguida, são descritas as conclusões e sugestões de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 GERENCIAMENTO DE PROJETOS

De acordo com o Guia PMBOK (2017), o gerenciamento de projetos busca alcançar a eficiência nas entregas de projetos em desenvolvimento nas mais diversas áreas. É responsável por aumentar a produtividade, impedir que prazos sejam estourados, otimizar o uso de recursos, minimizar ou impedir retrabalho, entre muitas outras coisas. O gerenciamento de projetos nas, mais diversas áreas, almeja alcançar eficiência e agilidade de maneira a suprir os objetivos propostos.

A necessidade da gestão de projetos data de mais de 50 anos atrás. Segundo Harold Kerzner (2015), em projetos grandes, que utilizam tecnologias previsíveis, os métodos de gerenciamento de projetos tradicionais ocupavam um lugar importante. Em contrapartida, no atual ambiente empresarial, projetos com tamanha previsibilidade ocupam uma pequena parcela dos desenvolvimentos necessários para o crescimento e diferenciação da empresa. Projetos de engenharia de *software*, por exemplo, possuem a necessidade de mudanças recorrentes, o que não é contemplado completamente em metodologias tradicionais.

Kerzner (2015) comenta ainda que, devido à grande exposição às tecnologias e ferramentas virtuais que nasceram graças à internet, foram criadas novas possibilidades de trabalho, aproximando cada vez mais as equipes e apresentando novas oportunidades de atuações na gerência de projetos.

2.2 METODOLOGIAS LEVES

As metodologias leves são uma resposta à necessidade crescente de técnicas de gestão mais adaptativas, chamadas também de metodologias ágeis. Com uma quantidade muito menor de documentações, a equipe pode focar em desenvolver o projeto e principalmente em proporcionar valor para o cliente em cada entrega incremental dele. Muitos dos especialistas em metodologias tradicionais (ou pesadas) se mostraram resistentes às metodologias ágeis (Fowler, 2021). Kerzner (2015) explica que, diferentemente das metodologias tradicionais, as metodologias ágeis propõem a construção do projeto através de um estilo de comunicação informal, com poucas regras, práticas e documentações.

2.2.1 Scrum

Beck *et al* (2001) explicam como no crescente cenário tecnológico e surgimento de novas práticas de gestão de projetos, a metodologia Scrum adquiriu muitos adeptos ao redor do mundo, tendo como um dos seus pilares a priorização da comunicação e diminuição da quantidade de documentações e sua organização em blocos de tempo bem definidos para cada atividade.

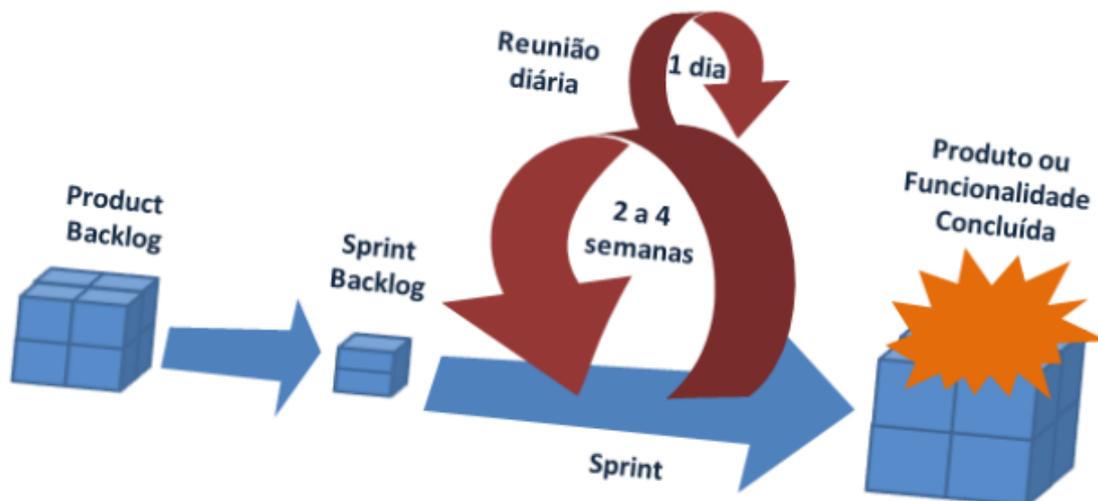
Na Figura 1 são apresentados os principais artefatos da metodologia Scrum. Em ordem cronológica, a primeira etapa é a estruturação do *Backlog* do Produto (*Product Backlog*, lista priorizada das atividades), em que é realizada a priorização das atividades pelo *Product Owner* (ou Dono do Produto, responsável por entender do negócio e funcionalidades). Conforme determinado no guia Scrum, por Schwaber e Sutherland (2020), o papel do *Product Owner* é também auxiliar o time Scrum no aumento do valor das entregas.

Em seguida, ocorre a estruturação do *Backlog* da *Sprint* (*Sprint Backlog*), que contém as atividades a serem realizadas durante a *sprint*. As

sprints são os períodos de tempo pré-definidos em que acontece o desenvolvimento do projeto – seu tamanho geralmente é de duas a quatro semanas.

As reuniões diárias são realizadas com o time de desenvolvimento diariamente e é uma das cerimônias da metodologia. A condução dessas reuniões é feita pelo *Scrum Master*. De acordo com Schwaber e Sutherland (2020), o *Scrum Master* é a pessoa responsável por fazer com que o time inteiro siga as regras da metodologia *Scrum*, auxiliando no cumprimento das entregas de valor e servindo tanto o time *Scrum* quanto o *Product Owner* (auxiliando com o gerenciamento do *backlog*, etc.) e a empresa (auxiliando na adoção da metodologia *Scrum*, removendo barreiras entre o time e os interessados.). As rodadas de *sprints* são realizadas até que o produto ou funcionalidade sejam concluídos e entregues aos interessados (*stakeholders*).

Figura 1 - Principais artefatos do Scrum



Fonte: Artigo no Medium: Método SCRUM — Um resumo de tudo o que você precisa saber.¹

Os projetos são divididos em entregas que buscam trazer valor para o solicitante do projeto. As equipes de desenvolvedores e gestores envolvidos no processo ganham denominações e atividades específicas, além de existir reuniões destinadas à cada etapa do projeto.

¹ Disponível em: <https://medium.com/reprogramabr/scrum-um-breve-resumo-f051e1bc06d9>. Acesso em 24 de maio de 2022.

2.2.1.1 A metodologia

A metodologia Scrum apresenta uma estrutura bem definida. Inicialmente o solicitante (ou cliente) traz o problema a ser solucionado pelo projeto. Dessa forma, é definido o escopo do desenvolvimento. As histórias, como são chamadas as funcionalidades macro, são então quebradas em entregáveis menores que irão compor o *backlog* do produto. Ao finalizar o *backlog* do produto, é realizada uma priorização para definir quais atividades serão realizadas primeiro. Dessa forma, é montado o *backlog* da *sprint*.

2.2.1.2 A Equipe

2.2.1.2.1 Desenvolvedores

Na metodologia Scrum, os desenvolvedores são o time que irá realizar as etapas de desenvolvimento do projeto. Diferentemente de outras metodologias, no Scrum, a equipe de desenvolvedores possui mais liberdade na tomada de decisões que surgem no desenrolar do projeto.

2.2.1.2.2 Scrum Master

O *Scrum Master* é o papel do gestor de projetos ao acompanhar o desenvolvimento e estar disponível para auxiliar com a resolução de problemas e dúvidas. Auxilia e direciona as reuniões diárias e de planejamento e fechamento de *sprint*, além de atuar como facilitador da comunicação dos membros envolvidos no projeto.

2.2.1.2.3 Product Owner

Esta é a pessoa responsável por definir a direção do projeto de acordo com os objetivos propostos pelos *Stakeholders*. O *Product Owner (PO)* é responsável pela priorização do *backlog*, escolhendo as atividades com base na sua importância para a entrega/projeto. O PO está em constante contato com os *Stakeholders*, visto que precisa entender a demanda e auxiliar na quebra de requisitos a partir das histórias.

O PO realiza a quebra das histórias em entregáveis que irão, no final de cada *sprint*, incrementar o projeto final. Ele também entende do negócio ao qual o projeto se refere e dessa forma pode auxiliar na resolução de dúvidas e questões que surgem durante o desenvolvimento.

2.2.1.3 Sprints

No Scrum, as entregas são realizadas ao final de períodos de tempos definidos. Esses períodos de tempo recebem o nome de *Sprints*. Schwaber e Sutherland (2020) apresentam as *sprints* como o batimento cardíaco do Scrum, e ainda afirmam que as *sprints* são onde as ideias se transformam em valor.

Mas de forma mais técnica, no planejamento do projeto é definido de quanto em quanto tempo serão realizadas as entregas e também fica determinado quais atividades precisam ser entregues no final da *sprint*, sendo esse, seu objetivo maior. A equipe precisa respeitar o tempo definido, que, comumente, é de duas semanas.

Schwaber e Sutherland (2020) descrevem que no decorrer da *sprint* não são realizadas mudanças que possam atrapalhar o objetivo principal dela. O *backlog* do produto pode ser redefinido conforme necessário e também que pode haver renegociações com o PO conforme o desenvolvimento avança.

2.2.1.4 Reuniões

A comunicação é chave quando se fala de metodologias ágeis. No Guia *Scrum*, Schwaber e Sutherland (2020) determinam que existem reuniões bem definidas para cada momento da *sprint* e que esses eventos são as ocasiões para que haja uma inspeção e adaptação da aplicação da metodologia. Portanto, a equipe se reúne de forma a garantir a transparência necessária para o projeto.

2.2.1.4.1 Planejamento da Sprint

No planejamento da *sprint* são priorizadas quais atividades serão realizadas no período determinado para a *sprint*. Essas atividades serão desenvolvidas durante a *sprint* e se tornarão entregas no final do processo. A reunião geralmente possui 2 horas de duração e envolve o PO e os desenvolvedores.

2.2.1.4.2 Reuniões diárias (dailys)

Um dos conceitos mais disseminados do Scrum são as reuniões diárias, conhecidas como *dailys*. Essas são, como o próprio nome diz, reuniões que acontecem diariamente. Quando o trabalho é realizado presencialmente, é comum que as pessoas envolvidas na reunião realizem a mesma em pé, para garantir agilidade. As *dailys* possuem o tempo fixo de 15 minutos e cabe ao *Scrum Master* dirigir a reunião e garantir que o foco não seja perdido.

Nesses pontos diários, cada membro da equipe de desenvolvedores, por sua vez, comenta o que fez pelo projeto no dia anterior, o que está fazendo hoje e o que irá fazer no próximo dia. Essas reuniões costumam acontecer pela

manhã para que os membros possam sair dela e já iniciar as discussões pertinentes ao desenvolvimento, tirando dúvidas, dando sugestões, auxiliando outros membros etc.

2.2.1.4.3 Reunião de fechamento da *sprint*

Ao final de toda *sprint* é realizado o seu fechamento. São reuniões com o tempo geralmente definido de 2 horas com o objetivo de apresentar as entregas realizadas pelo time de desenvolvimento. Participam desta reunião o PO, o SM e os *stakeholders*.

2.2.1.4.4 Retrospectiva da *sprint*

Na retrospectiva da *Sprint*, a equipe se reúne e são expostas as dificuldades encontradas durante o desenvolvimento e alterações que possam ter ocorrido. Caso tenha ocorrido algum atraso, a equipe conversa para entender o motivo deste e como melhorar esse aspecto na próxima *sprint*. Essas reuniões existem para melhorar o trabalho da equipe em conjunto.

2.3 METODOLOGIAS TRADICIONAIS

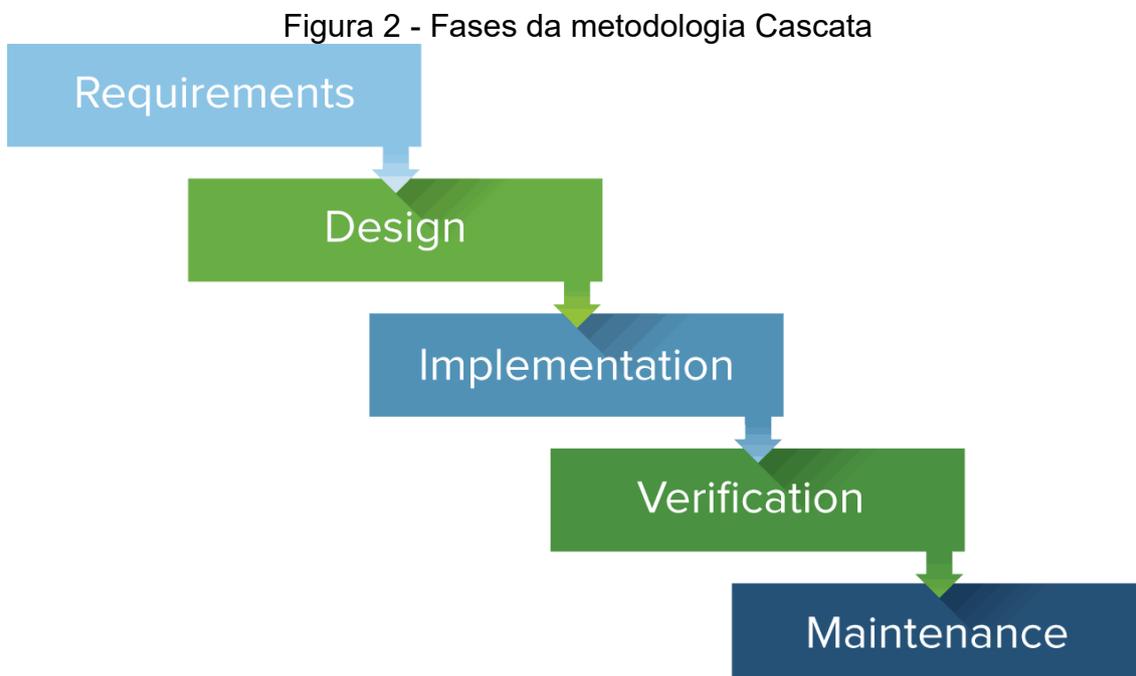
Para a construção de uma ponte, é necessário extensa preparação, documentação e previsibilidade, pois, é um projeto que no geral não sofre grandes alterações durante sua construção. As metodologias tradicionais nasceram para atender projetos como o de uma ponte, em que é possível prever cada etapa do processo de desenvolvimento. Kerzner (2015) pontua que tais metodologias estão perdendo cada vez mais a popularidade, pois a

quantidade de trabalho necessária para aplicação dela pode tornar o processo lento e resultar em retrabalhos, uma vez que não são preparadas para mudanças rápidas.

2.3.1 Método Cascata

O método cascata é um método de gerenciamento tradicional. Segundo Ahmad *et al* (2016), nele, as atividades são organizadas em uma sequência em que cada uma depende da anterior. É comentada também a forma como metodologias tradicionais como o método cascata buscam apresentar uma maior previsibilidade dos projetos, diferentemente das metodologias ágeis.

Em projetos que seguem o método cascata, existe um escopo do projeto bem definido desde o início, com etapas bem programadas e documentadas e um orçamento estipulado. Conforme mostrado na Figura 2.



Fonte: Publicação The Phases of Waterfall Methodology do site smartsheet.²

² Disponível em: <https://www.smartsheet.com/content-center/best-practices/project-management/project-management-guide/waterfall-methodology>. Acessado em 24 de maio de 2022.

Ahmed *et al* (2016) evidenciam que métodos tradicionais como o método cascata e o espiral são adequados a projetos que não apresentam tantas alterações durante seu desenvolvimento – por exemplo da área de construção civil. Entretanto, no mundo atual, com toda sua dinâmica e complexidade, muitas vezes não se adequam às necessidades do mercado quando se diz respeito à projetos de software onde nem sempre o escopo é bem definido e muitas mudanças podem ocorrer durante o projeto. Para muitas empresas, é necessário que alguns pontos do projeto sejam bem definidos. Quando ocorre a contratação de um projeto, é esperado que exista um orçamento definido, que pode até sofrer alterações, mas sem margens muito grandes.

É esperado também um período de desenvolvimento minimamente previsível, pois, o atual mercado de trabalho funciona com prazos finais para seus projetos. Nesse sentido, alguns aspectos do método cascata continuam existindo e sendo aplicados no ambiente empresarial. No entanto, já não é esperado que o desenvolvimento de um projeto ocorra sem nenhuma alteração.

2.4 KANBAN

No gerenciamento de projetos, Mali e Drury-Grogan (2021) comentam a forma como a metodologia Kanban é utilizada com a finalidade de melhorar na visualização do fluxo de trabalho. O Kanban é uma ferramenta famosa e amplamente utilizada quando se fala de metodologias ágeis, pois possibilita um acompanhamento visual do desenvolvimento do projeto e suas etapas. Anderson (2010) comenta que é possível destacar dois objetivos principais na aplicação da metodologia Kanban, a aplicação de processos de otimização específicos para cada contexto com pouquíssima resistência à mudanças no projeto e um ritmo de trabalho aceitável para os envolvidos no desenvolvimento.

O Kanban pode ser utilizado de diversas maneiras, tanto em meio físico quanto digital. É composto por colunas com os estados de desenvolvimento

como ilustrado na Figura 3, em que os estados são “a fazer”, “fazendo” e “feito”. Na coluna dos estados são adicionadas as atividades a serem desenvolvidas para o projeto em cartões, conforme uma atividade passa de um estado para outro, o cartão é movido entre as colunas para representar o estado real.

Figura 3 - Exemplo de Quadro Kanban



Fonte: Publicação “Você sabe o que é Kaban?” do site Reflow.³

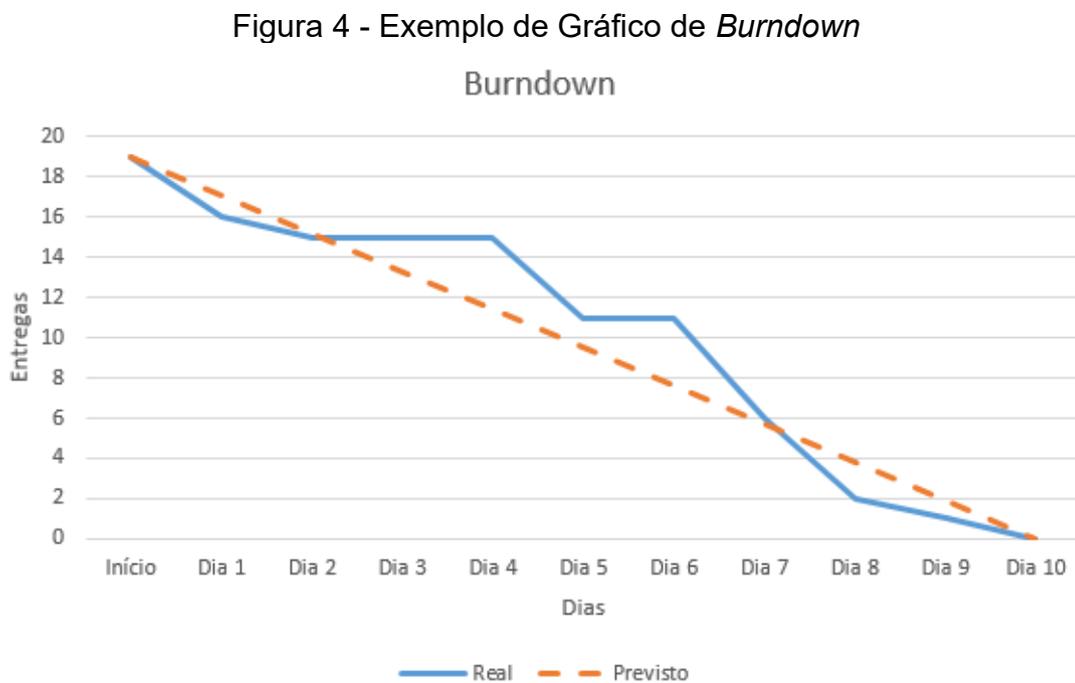
O Kanban traz uma maior visibilidade do estado atual do projeto para os envolvidos e interessados no mesmo. É comumente utilizado juntamente com metodologias ágeis para auxiliar no acompanhamento das atividades.

2.5 GRÁFICO DE BURNDOWN

Uma ferramenta utilizada nas reuniões diárias (*daily*s) é o gráfico de *burndown*. Com ele, é possível ter uma visualização gráfica das entregas versus tempo decorrido na *sprint*. Ele possibilita que seja observado rapidamente onde houve lacunas de entregas para, posteriormente, entender

³ Disponível em: <<https://www.reflow.com.br/post/voc%C3%AA-sabe-o-que-%C3%A9-kanban>>. Acessado em 24 de maio de 2022.

os motivos daquele ocorrido. Na Figura 4 é possível observar um modelo do gráfico de *burndown*.



Fonte: Autora.

Na figura 4 o eixo y é composto pelas entregas que devem ser realizadas diariamente e o eixo x pelos dias da *sprint*. A linha tracejada em laranja mostra o previsto de entregas e a linha contínua em azul descreve as entregas reais. Os pontos em que a linha azul está abaixo da laranjada mostram pontos em que as entregas estavam adiantadas com relação ao previsto. Já nos pontos em que a linha azul está acima da laranjada é possível ver onde as entregas estavam atrasadas em relação ao previsto.

3 METODOLOGIA

O projeto se trata de um estudo de caso. A análise e observação da gestão realizada durante o desenvolvimento de um projeto de software, utilizando um híbrido de metodologias ágeis e tradicionais, além de ferramentas de gestão pertinentes (*Kanban*). A pesquisa e observação é realizada na Empresa X.

É realizada fundamentação teórica dos temas pertinentes à pesquisa, utilizando artigos e livros relacionados que possibilitam uma melhor análise dos resultados encontrados, chegando assim a algumas conclusões.

O estudo utiliza da análise do fenômeno em seu contexto real, para entender como as metodologias aplicadas auxiliam na melhor obtenção de resultados pela equipe e na resolução de possíveis pendências encontradas no desenvolvimento.

4 DESENVOLVIMENTO DO PROJETO (ESTUDO DE CASO)

4.1 ANÁLISE INICIAL

Como dito a priori, o trabalho em questão se propõe a analisar o desenvolvimento de um projeto utilizando metodologias ágeis em conjunto com metodologias tradicionais, bem como analisar a produtividade da equipe durante o processo.

O projeto a ser analisado é de desenvolvimento do protótipo de um software, abrangendo as frentes de *front-end* (*desenvolvimento da parte visível da aplicação*), *back-end* (desenvolvimento da parte do código que faz com que a parte visível funcione corretamente, criando a ponte entre a interface e o banco de dados) e banco de dados da aplicação web. O intuito do projeto, bem como informações sensíveis referentes ao cliente ou aos desenvolvedores não serão apresentados. Serão utilizados apenas nomes fictícios. Portanto, o projeto será descrito como Projeto Z.

A Figura 5 representa uma simplificação do fluxo utilizado pela empresa X durante o processo de contratação de novos projetos. A empresa recebe pessoas interessadas tanto por prospecção ativa (ligações, etc.) quanto por prospecção passiva (indicações, etc.). Após a apresentação da empresa, é realizada a negociação e envio da proposta para o possível cliente. Em seguida, após negociações, é definido se será realizada a contratação do projeto ou não.

Figura 5 - Processo de contratação de novos projetos.



Fonte: Autora.

Existem dois tipos de contratação mais comuns utilizados na Empresa X. A primeira se caracteriza por horas por mês, em que o cliente assina um contrato relativo a uma quantidade predefinida de horas por mês e são disponibilizados recursos para atender as solicitações da empresa (geralmente para realizar pequenos desenvolvimentos). Nesse caso, podem ser realizados mais de um projeto por vez em mais de uma área da empresa. Esse meio de contratação é conhecido como Sustentação.

Já o outro método se caracteriza pela contratação por escopo fechado, em que, através das solicitações do cliente, é realizada uma estimativa inicial do esforço necessário para o desenvolvimento da atividade pela equipe contratada. A partir dessa estimativa, é construída a proposta para envio ao cliente, que define se realizará ou não a contratação.

O projeto analisado em questão foi contratado com escopo fechado. Esse método faz com que os clientes tenham ciência do que será gasto no desenvolvimento do projeto, desde que não ocorram grandes alterações.

A aplicação de metodologias ágeis em projetos de escopo fechado pode se mostrar um desafio, considerando que uma das premissas do desenvolvimento ágil é não limitar a equipe desenvolvedora. Entretanto, para muitos clientes, a ideia de um projeto sem início e final pré-definido é financeiramente inviável.

4.2 CRONOGRAMA

Na empresa X, quando são realizadas contratações com o escopo fechado, é muito comum o envio de um cronograma em conjunto com a proposta. Nesse cronograma, o cliente pode ver a previsão de entregas referentes a partes macros do desenvolvimento, além de datas de quando essas entregas serão realizadas.

A criação de tais cronogramas é realizada em uma aplicação de desenvolvimento próprio da empresa. Na Figura 6, é possível ver a interface de criação de um cronograma nesta aplicação.

Figura 6 - Interface do aplicativo de criação de cronogramas

Atividade	Pred.	Esforço	Duração	Início	Término	Recurso	Pap
0 Projeto Z			0 hr				
1 1 Conceção			0 hr	16 hrs	Sat 1/7/19	Sun 4/7/17	
18 2 Planejamento, Gestão e Ambiente			0 hr	0 hr	Sat 1/7/19	Sat 1/7/19	
25 3 Construção			0 hr	16 hrs	Sat 1/7/19	Sun 4/7/17	
119 4 Finalização	118	0 hr	16 hrs	Sat 1/7/19	Sun 4/7/17		

Fonte: A autora.

Antes da contratação, são realizadas reuniões entre o PO, o cliente (ou equipe de clientes) e a equipe comercial da empresa X com o objetivo de definir quais serão os tópicos macro a serem desenvolvidos. Posteriormente, esses tópicos são estimados em horas para construção do cronograma e proposta. Nesse desenvolvimento, os macros foram definidos conforme a Figura 7, considerando a construção de cinco telas e dois modais. Em cada requisito de tela ou modal, são consideradas as horas de *front-end* e *back-end* e gerenciamento, enquanto nos requisitos não funcionais é considerada a construção do banco de dados correspondente.

Figura 7 - Cronograma Inicial do Projeto Z

Id	Nome da tarefa	Esforço
0	Projeto Z - Cronograma	1284 hrs
1	1 Construção	1284 hrs
2	1.1 Entrega - Banco de Dados - Requisitos Não Funcionais	200 hrs
3	1.1.1 Marco - Desenvolvimento 01	200 hrs
4	1.1.1.1 RNF01 - Desenho de Estrutura do Banco e Desenho do Modelo DER	64 hrs
6	1.1.1.2 RNF02 - Validação do modelo e Estrutura de Dados	36 hrs
8	1.1.1.3 RNF03 - Criação da Estrutura DevOps	100 hrs
10	1.2 Entrega - Backend e Frontend - Requisitos Funcionais	0 hr
11	1.2.1 Marco - Desenvolvimento 01	804 hrs
12	1.2.1.1 RF01-Tela 1	140 hrs
14	1.2.1.2 RF02-Tela 2	160 hrs
16	1.2.1.3 RF03-Tela 3	140 hrs
18	1.2.1.4 RF04-Tela 4	120 hrs
20	1.2.1.5 RF05-Tela 5	150 hrs
22	1.2.1.6 RF06-Modal 1	46 hrs
24	1.2.1.7 RF07-Modal 2	48 hrs
26	1.3 Marco - Implementação 02	280 hrs
27	1.3.1 Documentação	20 hrs
29	1.3.2 Testes	160 hrs
30	1.3.2.1 Testes unitários back-end	80 hrs
31	1.3.2.2 Testes de Sistema	80 hrs
32	1.3.3 Homologação	100 hrs
33	1.3.3.1 Homologação do Cliente	100 hrs

Fonte: A Autora.

Após o envio da proposta, a quantidade de horas foi determinada como 1284. A duração total do desenvolvimento é calculada com base na quantidade de desenvolvedores disponíveis, levando em consideração quantas horas por

dia iriam trabalhar no projeto; A questão de duração do projeto será abordada melhor no tópico “Organização do *Backlog*”.

Com a contratação, o desenvolvimento pôde ser iniciado.

4.3 MONTANDO A EQUIPE

O desenvolvimento de uma aplicação web envolve muitas áreas de desenvolvimento. No caso do Projeto Z, a equipe contou com a participação de um desenvolvedor de *back-end* para realizar a programação por trás da aplicação, um desenvolvedor *front-end* para a programação da interface visual, um desenvolvedor de banco de dados e um desenvolvedor *fullstack* (desenvolvedor que programa tanto em *back-end* quanto em *front-end*) para auxiliar os outros membros da equipe quando necessário.

Além da equipe de desenvolvimento, participaram da realização do projeto também o PO que está em constante contato com o Cliente e o SM. A equipe Scrum foi estruturada, então, conforme o Quadro 1.

Quadro 1 - Estrutura da equipe

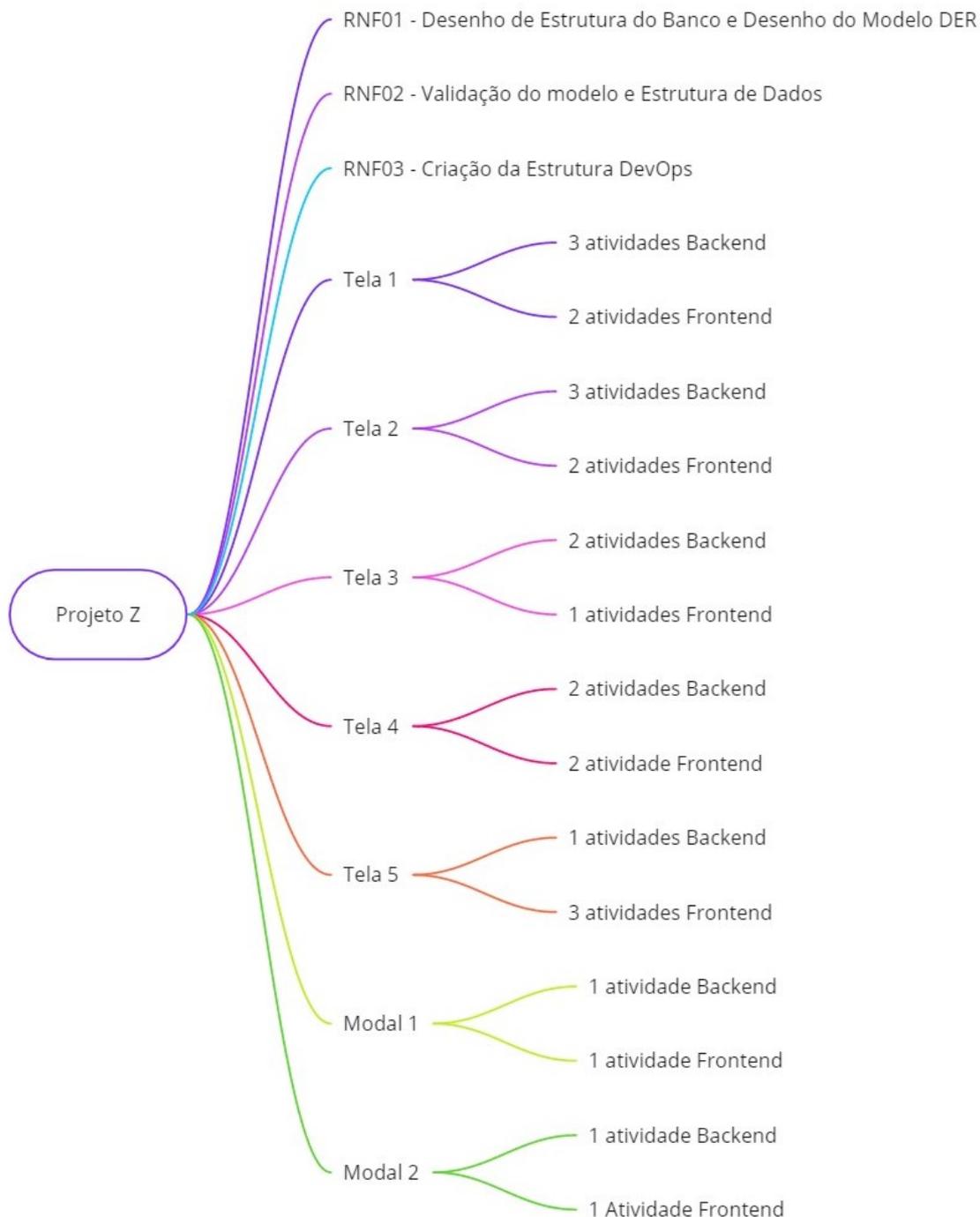
Membro	Papel
ProductOwner	PO
Scrum Master	SM
Equipe	D1 – Front-end D2 – Back-end D3 – Banco de dados D4 - Fullstack

Fonte: A autora.

4.4 INÍCIO DO DESENVOLVIMENTO

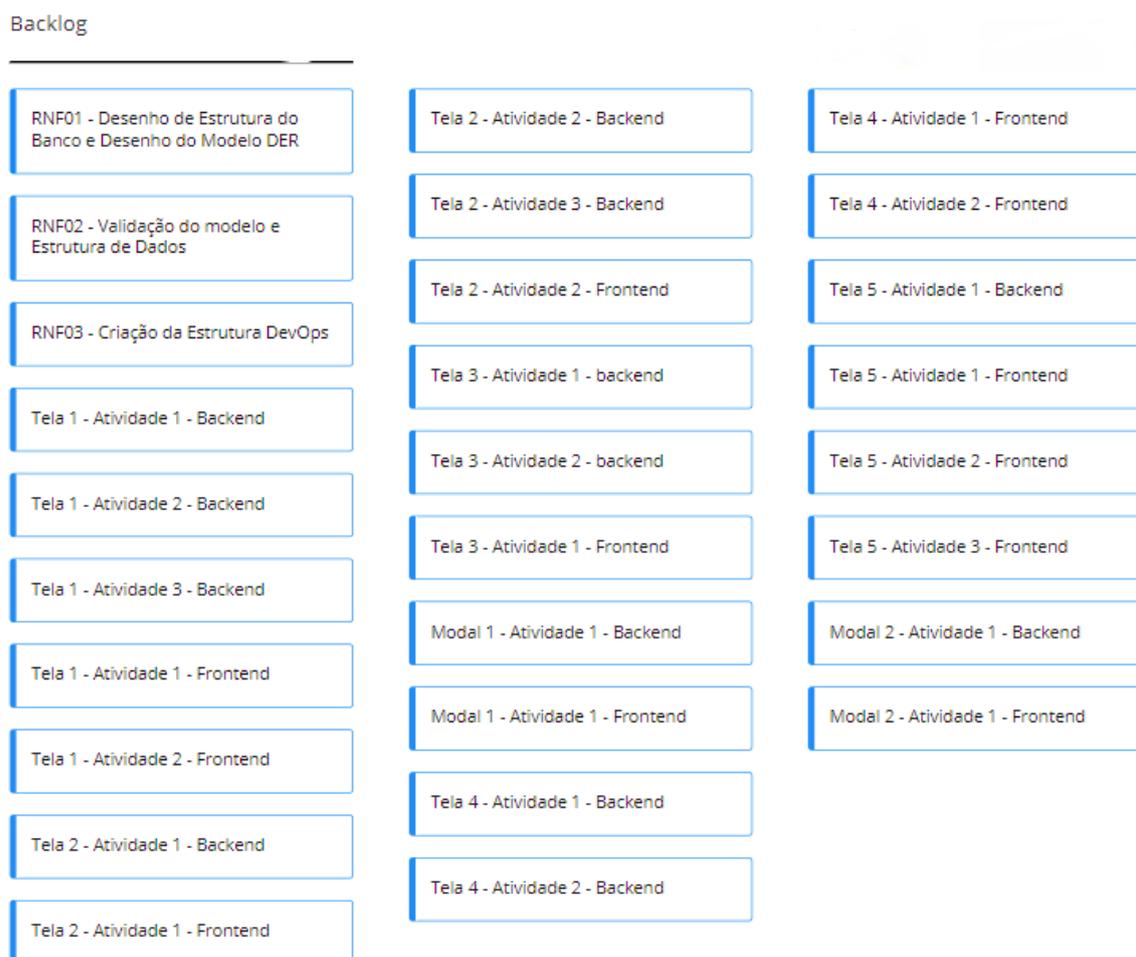
4.4.1 Quebra de atividades para *Backlog* do Produto

Após fechada a contratação, é iniciada a organização para início do projeto. A primeira etapa a se pensar é a quebra dos requisitos em atividades menores para compor o *backlog* do projeto, conforme pode ser visto na Figura 8. Foi utilizado um mapa mental a fim de simplificar a visualização das atividades necessárias para concluir cada tela e requisito. Ao final, foram totalizadas 28 atividades.

Figura 8 - Mapa Mental da Quebra de atividades para o *Backlog* do Produto

Fonte: a autora.

Então, foi realizada a priorização do *backlog* pelo PO em conjunto com o cliente. Na Figura 9, é possível ver as atividades após priorização. Para criação do *backlog*, foi utilizada a ferramenta Miro, que possibilita a criação de quadros online, onde pessoas podem compartilhar informações em tempo real.

Figura 9 - *Backlog* do Produto Priorizado

Fonte: a autora.

Como é possível observar, a priorização das atividades não segue exatamente a ordem do cronograma proposto inicialmente, como seria em um projeto acompanhado pelo método cascata. Nesse momento, se inicia a parte agilista do desenvolvimento.

Para as próximas etapas, foi definido que seriam realizadas *sprints* de duas semanas. A quantidade de horas a serem realizadas por cada membro da equipe de desenvolvimento por *sprint* é mostrada na Quadro 2.

Quadro 2 - Horas por *sprint* da equipe

Membro	Horas por <i>sprint</i>
D1	80
D2	80

D3	80
D4	80

Fonte: a autora.

As horas do SM e PO são adaptadas conforme a necessidade.

4.4.2 *SprintPlanning*

Após realizada a priorização das atividades no *backlog*, foi agendada com o cliente, PO e SM a realização do Planejamento da *Sprint* (popularmente conhecido como *sprint planning*). Nessa reunião, são separadas quais atividades são realizadas na *sprint* inicial do projeto. Essas atividades são mostradas no Quadro 3.

Quadro 3 – Divisão das atividades da primeira *sprint* para equipe

Atividade	Desenvolvedor
RNF01 - Desenho de Estrutura do Banco e Desenho do Modelo DER (64 horas)	D3
RNF02 - Validação do modelo e Estrutura de Dados (36 horas)	D3 e D4
RNF03 - Criação da Estrutura DevOps (100 horas)	D4 e D2

Fonte: a autora.

Com o planejamento, as atividades são separadas em partes menores, que constituem entregas menores que serão feitas durante a semana. As atividades da primeira *sprint* foram divididas da seguinte forma:

- RNF01 - Desenho de Estrutura do Banco e Desenho do Modelo DER – Seis partes
- RNF02 - Validação do modelo e Estrutura de Dados – Três partes
- RNF03 - Criação da Estrutura DevOps – Dez partes

No total, as atividades da primeira *sprint* foram separadas em 19 partes que seriam entregues no período de duas semanas.

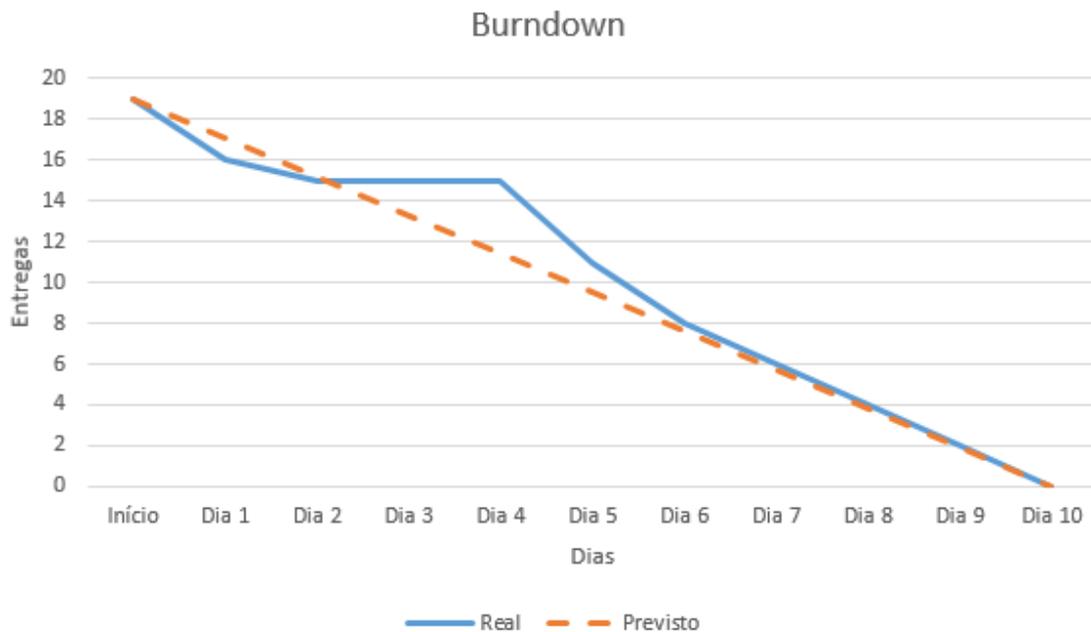
4.4.3 Reuniões Diárias (*Dailys*) da Primeira *Sprint*

As reuniões diárias ficaram agendadas para acontecer às 8:00 horas com a participação do SM e do time de desenvolvimento. O tempo para realização foi estipulado para a duração de 15 minutos, podendo exceder em casos extraordinários. Caso solicitado, a reunião poderia conter também o PO ou algum *Stakeholder*.

Nessas reuniões, os desenvolvedores são questionados sobre quais atividades realizaram no dia anterior, quais serão realizadas no dia e quais planejam realizar no próximo dia. São também questionados sobre possíveis dificuldades encontradas durante o desenvolvimento.

As reuniões são realizadas diariamente durante a *sprint* e com sua realização é montado o Gráfico de *Burndown*, que consiste em mostrar o desenvolvimento da equipe visualmente durante a *sprint* para que sejam observadas tendências e que ações devem ser tomadas para auxílio da equipe, caso necessário.

No final da primeira *sprint* a equipe entregou as 19 atividades propostas, que podem ser observadas no Gráfico de *Burndown* da Figura 10.

Figura 10 - Gráfico de *Burndown* da primeira *Sprint*

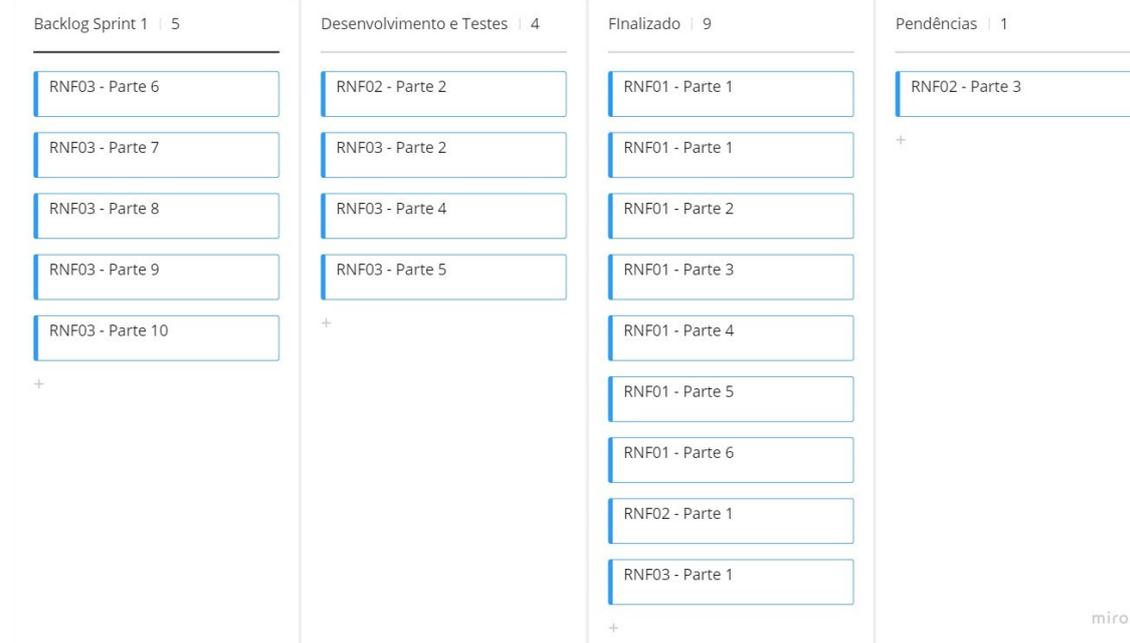
Fonte: a autora.

No gráfico, a linha tracejada em laranja é a linha de entregas ideal, prevista para o projeto na primeira *sprint*. Já a linha em azul mostra as entregas reais conforme foram realizadas pelos desenvolvedores nas reuniões diárias.

Pelo gráfico, é possível observar os pontos em que as entregas estão adiantadas – abaixo da reta prevista – e quando estão atrasadas com relação ao previsto – acima da reta prevista. Na *sprint* em questão, é observado que todas as entregas foram realizadas até o último dia da *sprint*, apesar de haver um certo atraso durante o desenvolvimento.

Além do gráfico de *burndown*, o acompanhamento também foi realizado através de um quadro Kanban, com as atividades da equipe. Na Figura 11 é mostrado como o Kanban se encontrava no sexto dia da *sprint*.

Figura 11 - Quadro Kanban da Primeira Sprint



Fonte: a autora.

Para se adaptar às realidades da equipe no momento, a coluna “A Fazer” foi renomeada como “Backlog Sprint 1”, a coluna “Fazendo” foi renomeada como “Desenvolvimento e Testes” e a coluna “Feito” foi renomeada como “Finalizado”. Além disso, foi adicionada uma coluna extra de pendências, em que as atividades que se encontravam com algum impedimento foram colocadas até o impedimento ser resolvido.

O Quadro Kanban foi utilizado diariamente nas *sprints* para que todos tivessem um panorama geral do momento de cada atividade.

Após cada reunião diária, os desenvolvedores atuaram nas atividades mais urgentes, resolvendo pendências e continuando o desenvolvimento. O SM pôde auxiliar na resolução de pendências também, nas situações em que elas dependiam do contato com o PO ou outras pessoas de fora da equipe.

4.4.4 Fechamento da *Sprint* e Retrospectiva

Outra forma encontrada de adaptação do Scrum para maior agilidade foi a realização consecutiva do fechamento e da retrospectiva da *sprint*. Na

retrospectiva, o time *Scrum* conversa a fim evitar que possíveis dificuldades se repitam. No caso da primeira *sprint*, foi discutido o atraso do Dia 2 ao Dia 6, para entendimento dos motivos e criação de estratégias futuras.

4.5 SPRINTS SEGUINTE

Para as *sprints* seguintes, foi planejado seguir o *backlog* conforme priorização, realizando os rituais da mesma forma que apresentado na Primeira *Sprint* do desenvolvimento. As *sprints* detalhadas não serão mostradas aqui para não estender muito o projeto e tornar sua leitura repetitiva. No entanto, a sequência determinada para realização das atividades seguiu conforme:

1. Planejamento da *Sprint*
2. Início do desenvolvimento
3. Reuniões diárias (utilizando Quadro *Kanban* e Gráfico de *Burndown*)
4. Fechamento e Retrospectiva da *Sprint*

4.6 DIFICULDADES ENFRENTADAS

Conforme o projeto se desenvolveu, nem todas as atividades foram realizadas conforme o tempo esperado, resultando em atrasos das atividades. Alguns impedimentos e problemas encontrados nesse momento serão expostos nos tópicos a seguir.

4.6.1 Estimativas incorretas

Durante o desenvolvimento das *sprints*, algumas estimativas de esforço se mostraram abaixo do esforço real necessário para realização de certas

atividades. Embora, o projeto tenha sido pré-combinado com o cliente, resumando uma quantidade específica de horas para produção de cada tela ou modal, algumas atividades se estenderam mais do que o esperado. Essa situação pode ocorrer tanto por falta de experiência da equipe para entendimento completo do problema proposto, ou mesmo pela falha de comunicação inicial para entendimento das histórias e requisitos.

O erro foi observado ao perceberem uma entrega em atraso constante nas atividades de *Back-end*, uma vez que os serviços estimados acabaram tomando um tempo mais elevado do desenvolvedor *Back-end*, acarretando posterior em atraso também das atividades de *Front-End*.

Para diminuir os atrasos apresentados, foi necessária introdução de mais um desenvolvedor *Full-stack*, com o intuito de auxiliar em pontos críticos do desenvolvimento para suprir esse tempo não estimado. A nova configuração da equipe pode ser observada no Quadro 4.

Quadro 4 - Configuração da Equipe após adição de recurso

Membro	Papel
ProductOwner	PO
Scrum Master	SM
Equipe	D1 – Front-end D2 – Back-end D3 – Banco de dados D4 – Fullstack D5 – Fullstack

Fonte: a autora.

Com a adição de um novo desenvolvedor, foi possível recuperar parcialmente o tempo de desenvolvimento.

4.6.2 Alteração de escopo

Conforme foram realizadas algumas entregas, os *Stakeholders*, em conjunto com o PO e a equipe de desenvolvimento, viram a necessidade de adicionar outras funcionalidades ao projeto. Com isso, foram necessárias realizar mudanças e alterações no escopo.

A Metodologia Scrum permite que tais medidas sejam tomadas e até facilita a alteração de pontos do projeto. A comunicação constante com os *Stakeholders* durante o desenvolvimento possibilita que possíveis falhas de projetos sejam percebidas e as mudanças sejam realizadas em tempo hábil.

De toda forma, alterações de escopo resultam em alterações de tempo de desenvolvimento e também de custos do projeto. No Projeto Z, foi realizado um novo cronograma para ser anexado ao anterior e solicitada nova proposta. O cronograma da mudança de escopo pode ser visto na Figura 12.

Figura 12 - Cronograma da Mudança de Escopo

Id	Nome da tarefa	Esforço
0	Projeto Z - Mudança de Escopo - Cronograma	534 hrs
1	1 Construção	534 hrs
2	1.1 Entrega - Backend e Frontend - Requisitos Funcionais	394 hrs
3	1.1.1 Marco - Desenvolvimento 01	394 hrs
4	1.1.1.1 RF08-Tela 6	140 hrs
6	1.1.1.2 RF09-Tela 7	160 hrs
8	1.1.1.3 RF10-Modal 3	46 hrs
10	1.1.1.4 RF11-Modal 4	48 hrs
12	1.2 Marco - Implementação 02	140 hrs
13	1.2.1 Documentação	20 hrs
15	1.2.2 Testes	80 hrs
16	1.2.2.1 Testes unitários back-end	40 hrs
17	1.2.2.2 Testes de Sistema	40 hrs
18	1.2.3 Homologação	40 hrs
19	1.2.3.1 Homologação do Cliente	40 hrs

Fonte: a autora.

A equipe comercial então enviou a nova proposta para o cliente e, após aprovação, foi possível seguir com o desenvolvimento das alterações.

As atividades macro de “Telas” ou modais foram quebradas em atividades menores e depois em partes para serem inseridas nas *sprints*.

4.6.3 Reorganização da equipe

Na realização das *sprints*, não foi possível manter a mesma equipe em todos os momentos. Houve reorganizações não esperadas e remanejamento de membros da equipe para outros projetos mais urgentes, o que acarretou em atrasos não programados.

No ambiente empresarial de tecnologia, é muito comum que haja remanejamentos por diversas causas. Um membro pode abandonar a equipe ou mesmo a empresa. Assim, recursos menos experientes são alocados para aquisição de conhecimento com desenvolvedores mais experientes.

No caso do Projeto Z, houve um atraso significativo devido a um recurso que foi solicitado por outra equipe por possuir conhecimento em um sistema desconhecido por todos os outros membros da empresa. Nesse momento, não foi possível alocar outra pessoa para a equipe, ocasionando em um atraso considerável.

Além disso, um desenvolvedor *Front-end* da equipe precisou dividir seu tempo com outros dois projetos, o que acabou dificultando a finalização de suas atividades dentro do tempo estipulado para as *sprints*.

Para tentar amenizar o atraso, alguns membros do time de desenvolvimento acabaram realizando horas extras.

4.6.4 Atrasos

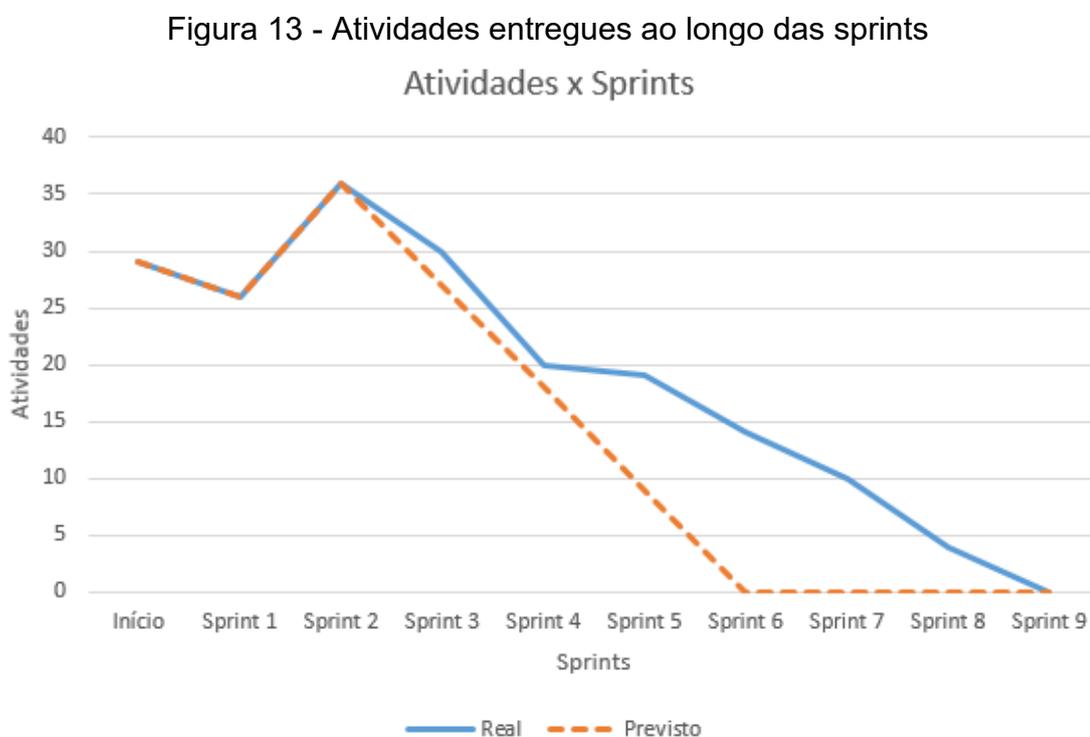
Além dos problemas maiores apresentados, também houve a incidência de atrasos fora do controle da equipe. Algumas atividades dependiam não só dos membros da equipe de desenvolvimento. Certas informações eram solicitadas aos *stakeholders* ou às pessoas de outras equipes e nem sempre eram retornadas com prontidão.

Nesses momentos, o SM se posicionava para realizar cobranças e tentar agilizar ao máximo a entrega das atividades solicitadas. A utilização da metodologia ágil nesse momento faz com que a necessidade da cobrança seja notada logo quando o atraso acontece e possibilita uma ação mais rápida.

4.6.5 Entrega do Projeto

Com o tempo total do projeto estimado no cronograma inicial, mais o cronograma da mudança de escopo, sendo contabilizado em 1818 horas totais, era esperado que o projeto fosse finalizado em média em 6 *sprints*, chegando em 3 meses de desenvolvimento. Entretanto o projeto foi entregue em 4 meses e meio, totalizando 9 *sprints*.

Na Figura 13 é possível observar como as entregas ocorreram durante as *sprints* de desenvolvimento em comparação com o inicialmente previsto para o projeto.



Na *sprint 2*, houve a adição das atividades que não estavam no escopo original, aumentando a quantidade de entregas a serem realizadas.

Enquanto em seguida, na *sprint 5* é possível observar o atraso devido aos erros de estimativa e também de remanejamento da equipe.

5 CONCLUSÕES

As metodologias ágeis têm presença constante na realidade empresarial, mas sua evolução implica que as empresas estejam sempre em adaptação para tentar aplicar suas ferramentas da forma que melhor se encaixe em suas realidades.

No projeto Z, foi utilizado um híbrido entre o método cascata e a metodologia Scrum, utilizando as ferramentas de ambas as metodologias. Na Empresa X, foi necessária a utilização da metodologia cascata para organização do projeto e criação de proposta, visto que é incomum a contratação de projetos sem um valor final acordado entre as partes interessadas.

A cultura da empresa ainda apresenta algumas dificuldades quanto à aplicação total de metodologias ágeis devido à grande rotatividade de membros das equipes. É importante destacar que, nem sempre, os recursos possuíam dedicação exclusiva a um mesmo projeto, o que pode atrapalhar o planejamento das *sprints* e possibilitar atrasos das entregas.

Durante o desenvolvimento, apareceram adversidades que logo puderam ser contornadas, pois foram identificadas as lacunas e atrasos assim que aconteciam. A estrutura da equipe possibilitou que o time de desenvolvimento tivesse disponível um suporte composto pelo SM e o PO, sempre prontos para identificar e solucionar pendências e realizar as devidas cobranças, quando necessário.

Apesar do atraso na entrega final, houve evidente melhora na comunicação da equipe de desenvolvimento e as informações percorriam um menor espaço de tempo até chegarem aos *stakeholders*, SM e PO. O projeto pode ser entregue, não totalmente como pensado no início, mas de forma a melhor atender às necessidades dos *stakeholders* e seus propósitos finais.

6 TRABALHOS FUTUROS

Como mencionado anteriormente, a aplicação de metodologias ágeis, em conjunto com metodologias mais tradicionais, ainda é uma saída quando se fala da adaptação de empresas aos métodos ágeis. Alguns futuros objetivos de estudo são descritos a seguir:

- Acompanhamento na melhora das entregas e dos métodos utilizados versus impacto que tiveram nos projetos;
- Criação e especificação de processos mais bem estruturados de gerenciamento de projetos dentro da empresa com intuito de dar suporte às equipes na aplicação das metodologias.

7 REFERÊNCIAS BIBLIOGRÁFICAS

KUHRMANN, Marco et al. **What makes agile software development agile**. IEEE transactions on software engineering, 2021.

AHMAD, Gul et al. An Overview: **Merits of Agile Project Management Over Traditional Project Management in Software Development**. Journal of Information & Communication Technology 10 (1), 105-120, 2016

BISHOP, David; ROWLAND, Pam; NOTEBOOM, Cherie. **Antecedents of preference for agile methods: A project manager perspective**. 2018. <https://doi.org/10.24251/HICSS.2018.678>

FORTE, Frank M.; KLOPPENBORG, Timothy. The Agile Mindset for Project Management. International Research Network on Organizing by Projects (IRNOP), 2018. <https://doi.org/10.5130/pmrp.irnop2017.5740>

DIEL, Elisa et al. **What is Agile, Which Practices are Used, and Which Skills are Necessary According to Brazilian Professionals: Findings of an Initial Survey**. 6th Brazilian Workshop on Agile Methods, 2015. <https://doi.org/10.1109/WBMA.2015.10>

CONBOY, Kieran; FITZGERALD, Brian. **Toward a conceptual framework of agile methods: a study of agility in different disciplines**. In: Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research. 2004. p. 37-44. <https://doi.org/10.1145/1029997.1030005>

BORGES, Carlos; ROLLIM, Fabiano. **Guia PMBOK® 6a. ed.** – EUA: Project Management Institute, 2017.

KERZNER, H. **Gestão de projetos: as melhores práticas**. Porto Alegre: Bookman, 2002.

M. Fowler, *The New Methodology*, Thought Works, 2001, Disponível em: <www.martinfowler.com/articles>.

BECK, Kent et Al. **Agile Manifesto web site**. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 20 jun. 2022.

SENAPATHI, Mali; DRURY-GROGAN, Meghann L. **Systems thinking approach to implementing kanban: A case study**. Journal of Software: Evolution and Process, v. 33, n. 4, p. e2322, 2021.
<https://doi.org/10.1002/smr.2322>

ANDERSON, David J. **Kanban: Successful Evolutionary Change for your Technology Business: Successful Evolutionary Change for your Technology Business**. Blue Hole Press; Blue Book ed. edição (7 abril 2010)

FICHA DE AVALIAÇÃO