

UNIVERSIDADE FEDERAL DE SANTA CATARINA

LUISA MAFRA DE OLIVEIRA

MODELO DE GERENCIAMENTO ÁGIL DE PROJETOS UTILIZANDO A  
METODOLOGIA KANBAN: aplicação em uma empresa de software

Joinville, SC  
2020

LUISA MAFRA DE OLIVEIRA

MODELO DE GERENCIAMENTO ÁGIL DE PROJETOS UTILIZANDO A  
METODOLOGIA KANBAN: aplicação em uma empresa de software

Trabalho de conclusão de curso a ser apresentado  
como requisito parcial para a obtenção do grau de  
Engenheiro de Transportes e Logística pela  
Universidade Federal de Santa Catarina - UFSC

Orientador: Prof. Cristiano Vasconcellos Ferreira, Dr. Eng.

Joinville, SC  
2020

LUISA MAFRA DE OLIVEIRA

MODELO DE GERENCIAMENTO ÁGIL DE PROJETOS UTILIZANDO A  
METODOLOGIA KANBAN: aplicação em uma empresa de software

Trabalho de conclusão de curso a ser apresentado  
como requisito parcial para a obtenção do grau de  
Engenheiro de Transportes e Logística pela  
Universidade Federal de Santa Catarina - UFSC

BANCA EXAMINADORA

---

Prof. Cristiano Vasconcellos Ferreira, Dr. Eng.  
Universidade Federal de Santa Catarina - UFSC

---

Prof. Carlos Maurício Sacchelli, Dr. Eng.  
Universidade Federal de Santa Catarina - UFSC

---

Prof. Janaina Renata Garcia, Dra.  
Universidade Federal de Santa Catarina - UFSC

## **AGRADECIMENTOS**

Em meio a todo esse período de pandemia que estamos vivendo, um capítulo muito especial na minha vida está sendo concluído. Agradeço a Deus pela oportunidade de viver o momento presente e expresse minha total gratidão por todo conhecimento e desenvolvimento pessoal e profissional que adquiri até aqui.

Aos meus pais, Claudia e Marcos e seus respectivos companheiros de vida, Luiz e Cristiane, o meu profundo agradecimento por me acompanharem durante toda a vida, sempre me dando muita força e as vezes uns puxões de orelha, obrigada por serem as minhas referências e porto seguro. Aos meus irmãos, Vitor, Leonardo e Nicolas agradeço por compartilharem essa vida comigo, que minha vitória lhes sirva para que nunca desistam dos seus sonhos. As minhas avós pelo exemplo e inspiração de vida, às minhas primas e primos por me trazerem tanta alegria e parceria, aos meus tios e tias por vibrarem tanto pela minha graduação e aos meus avôs que com certeza estão orgulhosos, mesmo não mais presentes fisicamente. Ao meu namorado Juliano, obrigada pelo apoio, incentivo e companheirismo durante essa fase importante da minha vida.

Não posso deixar de mencionar a importância do apoio das minhas amigas "prediletas" que estão comigo há tanto tempo e também ao meu grupo de amigos AAL, por tornarem Joinville e a faculdade uma conquista mais leve e divertida.

Agradeço imensamente ao meu orientador Dr. Cristiano Vasconcellos que me auxiliou pacientemente na condução deste trabalho, além de ter me acolhido tão bem durante minha jornada na universidade. Um especial agradecimento aos membros da banca, Dr. Carlos Sacchelli e Dra. Janaina Garcia por aceitarem o meu convite de fazer parte deste momento. Também gostaria de agradecer aos professores do CTJ pelo trabalho e dedicação ao ensino nesta universidade, meu reconhecimento e gratidão ao esforço, paciência e sabedoria que compartilharam comigo neste período.

Por fim, e não menos importante, agradeço à toda minha equipe da empresa, por contribuírem com meu desenvolvimento pessoal e profissional, ampliando meus conhecimentos a todo momento.

"I think goals should never be easy, they should force you to work, even if they are uncomfortable at the time." (Michael Phelps)

## RESUMO

Com a crescente competitividade em âmbito global e as mudanças nas demandas do mercado consumidor, há uma exigência que as empresas estejam em constante transformação no intuito de atender as necessidades dos clientes de forma eficaz e rápida. Por esse motivo, técnicas de melhorias e de otimização de processo começaram a ser mais desenvolvidas, especialmente no setor de Tecnologia. Nesse contexto, as metodologias ágeis ganharam espaço nos setores de desenvolvimento de software, principalmente por trazer uma organização nos processos, melhorar do fluxo de trabalho, reduzir o tempo de ciclo e aumentar o valor para o cliente. Entre as metodologias ágeis, uma das que mais se destaca é o Kanban. O Kanban em empresas de software oferece uma melhor visualização do fluxo de trabalho e com isso ajuda a identificar melhorias no processo de desenvolvimento. O objetivo do presente trabalho é propor um modelo de gerenciamento ágil de projetos utilizando a metodologia Kanban em uma empresa de software. O trabalho apresenta, inicialmente, uma revisão bibliográfica e um estudo de caso na empresa de software. E, na sequência, com base nestes resultados, o modelo proposto. E, ao final, a avaliação da empresa sobre o modelo proposto e um questionário de avaliação do modelo implementado na empresa.

**Palavras-chave:** Gerenciamento ágil de projetos. Planejamento do projeto de produtos. Empresas de software. Gerenciamento de Projeto. Kanban.

## **ABSTRACT**

With a growing increase in the global scope and changes in the demands of the consumer market, there is a recovery of companies that are constantly changing in order to meet the needs of customers effectively and quickly. For this reason, improvement techniques and process optimization are being implemented more recently, especially in the technology sector. In this context, as agile methodologies to gain space in the software development sectors, mainly for bringing a process organization, improving the workflow, reducing the cycle time and increasing the value for the customer. Among the agile methodologies, one of the most displayed in Kanban. Kanban in software companies offers a better response to the workflow and thereby helps to identify improvements in the development process. The objective of the present work is to propose a project management model that uses the Kanban methodology in a software company. The work presents, authorized, a bibliographic review and a case study in the software company. And then, based on these results, or on the proposed model. And, at the end, an evaluation of the company on the proposed model and a questionnaire to evaluate the model implemented in the company.

**Keywords:** Agile Project Management. Project product planning. Software companies. Project Management. Kanban

## LISTA DE ILUSTRAÇÕES

Figura 1 — O custo da mudança do escopo . . . . .	16
Figura 2 — Ciclo de gerenciamento de projeto . . . . .	19
Figura 3 — Objetivo da gestão de projeto – abordagem tradicional. . . . .	23
Figura 4 — Etapas modelo cascata . . . . .	24
Figura 5 — Valores do manifesto ágil . . . . .	25
Figura 6 — Princípios do manifesto ágil . . . . .	26
Figura 7 — Fluxo do processo scrum . . . . .	29
Figura 8 — Quadro Kanban . . . . .	33
Figura 9 — Quadro Kanban Trello . . . . .	34
Figura 10 — Fluxo de processos . . . . .	38
Figura 11 — Estrutura dos times de produto e engenharia na empresa. . . . .	44
Figura 12 — Tickets de email . . . . .	45
Figura 13 — Processo genérico de desenvolvimento das funcionalidades de um software . . . . .	46
Figura 14 — Estrutura do time estudado . . . . .	48
Figura 15 — Modelo de gerenciamento ágil de projetos baseados no Kanban. . . . .	50
Figura 16 — Divisão das entregas . . . . .	52
Figura 17 — Quadro Kanban utilizado no time estudado . . . . .	54
Figura 18 — Quantidade de itens na coluna "fazendo" do quadro kanban . . . . .	55
Figura 19 — Tempo de ciclo tempo total das tarefas . . . . .	57
Figura 20 — Diagrama de fluxo acumulado do projeto estudado . . . . .	59
Figura 21 — Throughput e Coeficiente de variação . . . . .	60
Figura 22 — Implementação de melhoria na quebra de tarefas por observação . . . . .	61



## LISTA DE QUADROS

Quadro 1 — Diferença da abordagem clássica e ágil de projetos . . . . .	14
Quadro 2 — Distribuição Setorial do Peso Médio dos Gastos em P&D no Faturamento e do Número de Empresas que os realizam. . . . .	15
Quadro 3 — Grupos de processo de gerenciamento de projetos e mapeamento das áreas de conhecimento do guia PMBoK . . . . .	20
Quadro 4 — Comparação entre GPA e GP "tradicional" . . . . .	27
Quadro 5 — Principais métodos ágeis . . . . .	27
Quadro 6 — Técnicas ágeis mais adotadas . . . . .	30
Quadro 7 — Scrum vs. Kanban . . . . .	36
Quadro 8 — Afirmações e suas respectivas médias . . . . .	61

## **LISTA DE ABREVIATURAS E SIGLAS**

CFD	Cumulative Flow Diagram
CoV	Coeficiente de variação
CRM	Customer relationship management
IP	Internet Protocol
PMBok	Project Management Body of Knowledge
SAAS	Software como serviço
TI	Tecnologia da informação
UFSC	Universidade Federal de Santa Catarina
WIP	Work in progress

## SUMÁRIO

1	<b>INTRODUÇÃO</b>	12
1.1	PROBLEMÁTICA	13
1.2	OBJETIVOS	14
1.3	JUSTIFICATIVA	15
1.4	ESTRUTURA DO TRABALHO	17
1.5	CONTEXTO NO CURSO	17
2	<b>FUNDAMENTAÇÃO TEÓRICA</b>	18
2.1	ABORDAGEM TRADICIONAL DE GERENCIAMENTO DE DESENVOLVIMENTO DE SOFTWARE	18
2.2	MÉTODOS ÁGEIS DE GERENCIAMENTO DE PROJETO	24
2.2.1	<b>Método Scrum</b>	28
2.2.2	<b>Método Kanban</b>	31
2.3	SOFTWARE	36
2.4	PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	37
2.4.1	<b>Levantamento de requisitos</b>	39
2.4.2	<b>Análise de Requisitos</b>	39
2.4.3	<b>Planejamento</b>	40
2.4.4	<b>Implementação</b>	40
2.4.5	<b>Testes</b>	40
2.4.6	<b>Manutenção</b>	40
3	<b>ESTUDO DE CASO</b>	42
3.1	CARACTERIZAÇÃO DA EMPRESA	42
3.2	O CASO DE ESTUDO	44
4	<b>MODELO DE GERENCIAMENTO ÁGIL DE PROJETO BASEADO NO KANBAN</b>	49
4.1	MAPEAMENTO DO PROBLEMA	50
4.2	DEFINIÇÃO DAS ONDAS DE ENTREGAS	51
4.3	DEFINIÇÃO DAS PRÁTICAS KANBAN A SEREM UTILIZADAS	52
4.3.1	<b>Modelagem e implementação do quadro kanban</b>	53
4.3.1.1	Limitação de quantidade de itens em progresso	54
4.3.2	<b>Implementação das cerimônias Kanban</b>	56
4.4	ANÁLISE DAS MÉTRICAS COLETADAS	56
4.4.1	<b>Tempo de ciclo</b>	56
4.4.2	<b>Diagrama de fluxo acumulado</b>	58
4.4.3	<b>Throughput e coeficiente de variação</b>	59
4.5	IMPLEMENTAÇÃO DE PEQUENAS MELHORIAS	60
		61

4.6	ANALISE DO QUESTIONÁRIO DE AVALIAÇÃO DA SOLUÇÃO PROPOSTA.....	61
5	<b>CONCLUSÃO</b> .....	64
	<b>REFERÊNCIAS</b> .....	66
	<b>APÊNDICE A — QUESTIONÁRIO DE AVALIAÇÃO DA SOLUÇÃO PROPOSTA</b> .....	70

## 1 INTRODUÇÃO

A crescente competitividade em âmbito global e as mudanças nas demandas do mercado consumidor, exigem que as empresas estejam em constante transformação no intuito de atender as necessidades dos clientes de forma eficaz e rápida. No contexto apresentado, um exemplo de segmento que está em contínua mudança e evolução como consequência de sua alta concorrência é a indústria de software.

Devido a abrangência que os sistemas de software hoje oferecem e, por isso, uma alta na complexidade, se faz necessário um processo de desenvolvimento do produto mais elaborado e detalhado com o objetivo de atender aos requisitos de um mercado consumidor cada vez mais exigente e instruído.

Este cenário impacta no crescimento das diferentes formas de gerenciamento de projetos na área de software. O gerenciamento de projetos pode ser definido como o planejamento, a programação e o controle de uma série de tarefas integradas à forma de atingir seus objetivos com êxito por meio do qual a empresa programa suas constantes entregas de serviços aos clientes, seja eles finais ou internos (KERZNER, 2007).

Projetos com a característica de inovação na área de software possuem alta complexidade, risco e a necessidade de ser fazer de forma veloz a acompanhar o mercado, portanto tornam-se críticos se não forem devidamente analisados e gerenciados. Segundo Goelzer et al. (2014), “projetos de inovação, além de exclusivos, ainda possuem uma maior probabilidade de riscos e elevados custos, por conta da sua complexidade e da imprevisibilidade”.

A partir de uma perspectiva de longo prazo, um projeto que gera uma aprendizagem sistemática do seu gerenciamento permite a empresa desenvolver competências de projeto que levam a uma vantagem competitiva sustentável Schindler e Eppler (2003). A documentação sistemática de dificuldades, erros ou armadilhas potenciais ajuda a reduzir os riscos do projeto (FERENHOF; FORCELLINI, 2011).

No ano de 2001, alguns profissionais reuniram-se para discutir suas experiências vivenciadas em gerenciamento de projetos e definiram um conjunto de valores e princípios que delinearão o que ficou conhecido como Manifesto Ágil Beck et al. (2001). O gerenciamento ágil de projetos é uma alternativa para o desenvolvimento de software onde são necessários não só o conhecimento técnico, mas também o compartilhamento de conhecimento implícito, trabalho em equipe e uma liderança atenta para os problemas a serem resolvidos.

Neste cenário, ocorreu o presente trabalho, o qual foi desenvolvido no âmbito

de uma empresa do segmento conhecido como SaaS (software como serviço) que tem um produto inovador na área de marketing digital para países emergentes. O escopo deste projeto envolveu o estudo e a otimização dos métodos de gerenciamento de projetos nas áreas de engenharia e produto, tendo como foco as técnicas e práticas de gerenciamento ágil de projeto baseado no Kanban.

O Kanban é um termo de origem japonesa que significa literalmente “cartão” ou “sinalização”. Tal conceito está diretamente relacionado com a utilização de cartões, para indicar o andamento dos fluxos de produção em empresas de fabricação em série.

O público alvo do trabalho foram as empresas e os profissionais que atuam na área de produto e engenharia da empresa, no que se refere ao incremento da velocidade do gestor em analisar estratégias para o desenvolvimento de software, permitindo que a empresa se mantenha no mercado que está sempre se atualizando e cada vez mais competitivo.

## 1.1 PROBLEMÁTICA

Quando uma empresa opta pela inovação, está, em maior ou menor grau, tomando uma decisão organizacional e assumindo riscos Goelzer et al. (2014). No mercado de software existe uma alta concorrência e grande necessidades de atualizações constantes. Portanto, projetos dentro deste mercado, tornam-se críticos se não forem devidamente analisados e gerenciados (PINTO; FRANK; PAULA, 2011).

Ainda hoje, na maioria das empresas é utilizado o modelo formal e tradicional de gerenciamento de projetos, amplamente difundido para seus processos de negócio e, principalmente, para o desenvolvimento de sistemas de tecnologia da informação (TI), conhecido como modelo em cascata de gerenciamento de projeto. Segundo Costa (2017), na abordagem tradicional, todo o processo de desenvolvimento de um software é esboçado e documentado em um momento anterior a implementação. Isto se faz necessário devido ao alto custos para realizar modificações no código devido às limitações tecnológicas da época.

Conforme a empresa inova no modo no qual os projetos são gerenciados, a mesma tende a se modernizar. Atualmente, uma inovação na forma de gerenciar projeto é a adoção da **metodologia ágil**, que prega a simplificação dos métodos atuais, o planejamento iterativo, a simplicidade, a flexibilidade e a autogestão (AMARAL et al., 2011).

Assim, com o decorrer do tempo, o processo de desenvolvimento de software passou a ser caracterizado por meio de rápidas e contínuas entregas de software

com valor agregado; mudanças de requisitos, inclusive em etapas avançadas do desenvolvimento; entregas de software funcionando frequentemente, em algumas semanas ou meses, de preferência no menor tempo possível; o software funcional é a medida primária de progresso.

O quadro 1 mostra as características do gerenciamento tradicional de projetos baseado no modelo do PMBoK (Project Management Body of Knowledge) do PMI (2013) e da abordagem ágil de projetos.

Quadro 1 - Diferença da abordagem clássica e ágil de projetos

	Clássica	Ágil
Desenvolvedor	Hábil	Ágil
Clientes	Pouco envolvido	Comprometido
Requisitos	Conhecidos e estáveis	Emergentes e mutáveis
Retrabalho	Caro	Barato
Planejamento	Direciona resultados	Resultados o direcionam
Foco	Grandes projetos	Projetos de natureza exploratória e inovadores
Objetivo	Controlar em busca de alcançar o planejado	Simplificar processo de desenvolvimento

Fonte: SHENHAR e DVIR (1996)

Desta forma, a grande questão a ser respondida é como as empresas da área de software podem aprimorar o seu processo de gerenciamento de projetos e desenvolvimento de software, indo ao encontro da abordagem da gestão ágil. A empresa na qual foi realizado este trabalho, utiliza o Kanban, entretanto, identificou-se a oportunidade de propor um modelo estruturado e detalhado, referente a utilização do Kanban, constituindo assim uma proposta de um modelo de gerenciamento ágil de projetos utilizando a metodologia Kanban.

## 1.2 OBJETIVOS

Este trabalho tem como objetivo geral propor um modelo de gerenciamento ágil de projetos utilizando a metodologia Kanban.

Neste contexto, constituem objetivos específicos do presente trabalho:

- Revisar bibliograficamente as abordagens clássicas e ágeis de gerenciamento de projetos;
- Identificar o modelo, práticas, técnicas e ferramentas de gestão do gerenciamento ágil;
- Compreender o processo de desenvolvimento de software, e suas

diferentes etapas; e,

- Adequar o modelo teórico através de um estudo de caso e avalia-lo.

### 1.3 JUSTIFICATIVA

Segundo a Fundação de Amparo à Pesquisa do Estado de São Paulo (2018), o Brasil forma aproximadamente 68 mil engenheiros por ano, porém segundo Ikeda (2012), apenas 13 de cada 100 engenheiros tem o perfil que o mercado espera. Ainda de acordo com o autor citado acima, em uma pesquisa realizada com médias e grandes empresas há deficiência na comunicação, em práticas na gestão de projetos e limitações em exercer liderança.

De acordo com Reis (2013), falta desenvolver na faculdade disciplinas que complemente a parte de gerenciamento de projetos e gestão, ele aponta que essa é uma falha estrutural da grade curricular. As teorias das abordagens dos métodos ágeis de gerenciamento de projetos têm ganhado força cada vez mais. A escolha adequada de uma metodologia de gerenciamento de projeto faz com que a organização seja mais eficiente em termos de planejamento, de definição de marcos no projeto e de melhoria da qualidade dos seus produtos. Sendo assim, as indústrias que desenvolvem software têm adotado de forma crescente práticas ágeis e o foco de pesquisa nessa área vem aumentando ao longo dos anos (DINGSØYR et al., 2012).

Portanto, esse trabalho se faz necessário por discutir e apresentar práticas em gerenciamento de projetos de uma forma mais atual utilizando-se da metodologia ágil, que possa ser utilizado como referência para muitos estudantes e profissionais que trabalham ou desejam trabalhar na área mais específica de gerenciamento de projetos ágil com foco em softwares.

Em 2018, o segmento de software representava a segunda maior indústria em distribuição dos gastos em pesquisa e desenvolvimento no mundo, segundo o quadro 2.

Quadro 2 - Distribuição Setorial do Peso Médio dos Gastos em P&D no Faturamento e do Número de Empresas que os realizam. (continua)

Indústria	P&D (% do faturamento)	Numero de empresas
Farmacêutica e Biotecnologia	15,7%	369
Software e Serviços de Computação	10,9%	278
Brinquedos	9,7%	2
Hardware e Equipamentos	8,7%	298
Serviços financeiros	6,0%	16



Quadro 2 - Distribuição Setorial do Peso Médio dos Gastos em P&D no Faturamento e do Número de Empresas que os realizam. (conclusão)

Indústria	P&D (% do faturamento)	Numero de empresas
Bens de lazer	6,0%	38
Equipamentos elétricos e eletrônicos	4,8%	228
Instrumentos de investimentos Equity	4,7%	1
Automóveis e autopeças	4,4%	156
Aeroespacial e Defesa	4,3%	53

Fonte: INSTITUTO DE ESTUDOS PARA O DESENVOLVIMENTO INDUSTRIAL (2018)

Durante o desenvolvimento de softwares erros no planejamento do projeto e documentação do software resultam em alterações complexas e de custos elevados. É possível notar que uma modificação na etapa final pode custar até 200 vezes o valor de uma modificação de requisitos, segundo a figura 1 que representa a piramide do custo da mudança de escopo.

Figura 1 - O custo da mudança do escopo



Fonte: Amancio et al. (2015)

Em suma, este cenário mostra a importância do estudo do tema **gestão ágil de projeto** aplicado no segmento **industrial de software**. Além disto, corroboram para a justificativa do trabalho, os inúmeros desafios do gerenciamento de projeto, amplamente difundidos na literatura e observados na prática, os quais envolvem mudança de escopo constante, estouro de orçamento, dificuldade de comunicação, atraso de lançamento de produtos, entre outros.

## 1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos.

No Capítulo 1 são apresentadas uma breve introdução sobre o tema a ser discutido, os objetivos gerais e específicos, as questões que levaram à realização deste trabalho e a estrutura escolhida para o trabalho.

O Capítulo 2 trata da fundamentação teórica. São apresentadas e analisadas as referências dentro do escopo deste trabalho, abordados os termos de engenharia de software, gerenciamento de projetos e processo e metodologias ágil.

Já o Capítulo 3 apresenta o estudo de caso em uma empresa do gerenciamento de um projeto em desenvolvimento na área de engenharia de software.

No Capítulo 4, é apresentada a proposta de implementação do processo de gerenciamento.

No Capítulo 5 são discutidas as lições aprendidas na aplicação do gerenciamento de projeto com metodologia ágil e as conclusões deste trabalho e apontamentos para trabalhos futuros.

## 1.5 CONTEXTO NO CURSO

Este documento apresenta o trabalho desenvolvido na disciplina EMB 5045 – Trabalho de Conclusão de Curso. O aluno irá utilizar conhecimentos aprendidos em matérias relacionadas a área de informática como: EMB 5600 - Programação I, EMB 5630 - Programação II, EMB 5631 - Programação III e EMB 5113 - Modelagem de Sistemas.

Além de matérias na área de gestão e projetos como: EMB 5120 - Gestão e Organização, EMB 5042 - Metodologia de Projeto de Produtos e EMB 5962 - Planejamento Estratégico.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para que um processo de desenvolvimento de software ocorra de forma adequada é necessário que o mesmo seja gerenciado. No capítulo 2, serão apresentadas duas metodologias em gerenciamento de projetos, a tradicional e a ágil. A apresentação destas metodologias terá como foco na indústria de software. Logo após, será contextualizado software e seu desenvolvimento genérico.

### 2.1 ABORDAGEM TRADICIONAL DE GERENCIAMENTO DE DESENVOLVIMENTO DE SOFTWARE

A abordagem tradicional do gerenciamento de desenvolvimento de software está baseada na abordagem de gerenciamento de projetos, descritas PMBoK (PMI, 2013). De acordo com este guia um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. A natureza temporária dos projetos indica que eles têm um início e um término definidos e podem criar:

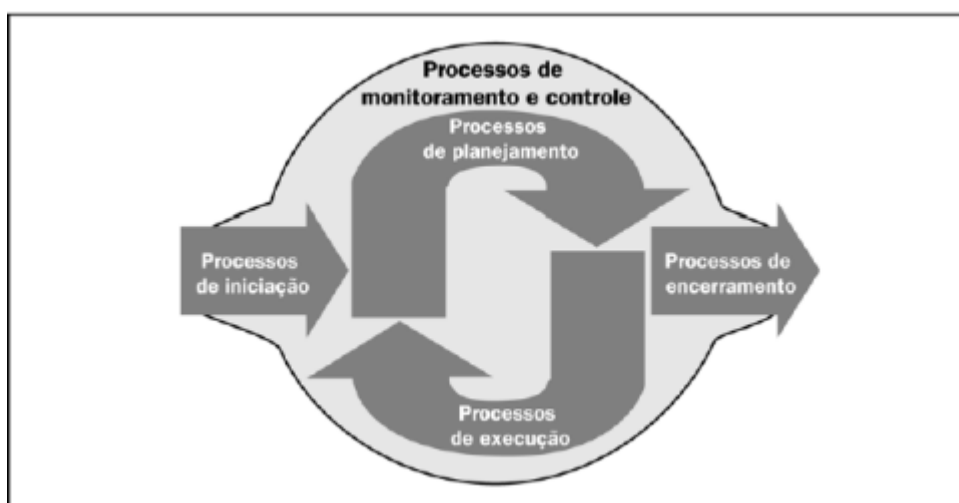
- Um produto que pode ser um componente de outro item, um aprimoramento de outro item, ou um item final;
- Um serviço ou a capacidade de realizar um serviço (p.ex., uma função de negócios que dá suporte à produção ou distribuição);
- Uma melhoria nas linhas de produtos e serviços (por exemplo, um projeto Seis Sigma executado para reduzir falhas); ou
- Um resultado, como um produto ou documento (por exemplo, um projeto de pesquisa que desenvolve o conhecimento que pode ser usado para determinar se uma tendência existe ou se um novo processo beneficiará a sociedade).

O guia PMBoK é um compilado de conhecimentos relacionados às melhores práticas, técnicas e teorias de gestão ligadas ao gerenciamento de projetos. Atividade onde é necessária a aplicação do conhecimento, habilidades, ferramentas e técnicas para atender aos requisitos exigidos pelo projeto. O gerenciamento de projeto pode ser entendido como sendo uma atividade onde é necessária a aplicação do conhecimento, habilidades, ferramentas e técnicas para atender aos requisitos exigidos pelo projeto. O gerenciamento de projetos é realizado por meio da aplicação e integração apropriadas dos 47 processos de gerenciamento de projetos, logicamente agrupados em cinco grupos de processos.

Os cinco grupos de processos descritos no PMBoK são: i) Processos de Iniciação que aprova e define o início do projeto ou fase; ii) Processos de

Planejamento aprimora os objetivos do projeto e define as ações; iii) Processos de Monitoramento e Controle como o próprio nome diz monitoram as atividades do projeto regularmente; iv) Processos de Execução aciona os recursos (humanos e materiais) necessários para a execução das atividades; v) Processos de Encerramento formaliza a aceitação do resultado da fase ou do projeto. Estes grupos se relacionam durante todo o processo de projeto, conforme ilustrado pela figura 2.

Figura 2 - Ciclo de gerenciamento de projeto



Fonte: PMI (2013)

O gerenciamento de um projeto normalmente inclui, mas não se limita a:

- Identificação dos requisitos;
- Abordagem das diferentes necessidades, preocupações e expectativas das partes interessadas no planejamento e execução do projeto;
- Estabelecimento, manutenção e execução de comunicações ativas, eficazes e colaborativas entre as partes interessadas;
- Gerenciamento das partes interessadas visando o atendimento aos requisitos do projeto e a criação das suas entregas;
- Equilíbrio das restrições conflitantes do projeto que incluem, mas não se limitam, a: Escopo, Qualidade, Cronograma, Orçamento, Recursos e Riscos.

Devido ao potencial de mudanças do projeto, o desenvolvimento do plano de gerenciamento do projeto é uma atividade iterativa elaborada de forma progressiva ao longo do ciclo de vida do projeto. A elaboração progressiva envolve a melhoria contínua e o detalhamento de um plano conforme informações mais detalhadas e específicas e estimativas mais exatas tornam-se disponíveis. A elaboração progressiva permite que a equipe de gerenciamento do projeto defina e gerencie o trabalho com um nível maior de detalhes, à medida que o projeto evolui.

O PMBoK PMI (2013), também está organizado em 10 áreas de conhecimentos. Uma área de conhecimento representa um conjunto completo de conceitos, termos e atividades que compõem um campo profissional, campo de gerenciamento de projetos, ou uma área de especialização. Essas dez áreas de conhecimento são usadas na maior parte dos projetos, na maioria das vezes. As equipes dos projetos utilizam essas e outras áreas de conhecimento, de modo apropriado, para os seus projetos específicos.

As áreas de conhecimento são: Gerenciamento da integração do projeto, Gerenciamento do escopo do projeto, Gerenciamento do tempo do projeto, Gerenciamento dos custos do projeto, Gerenciamento da qualidade do projeto, Gerenciamento dos recursos humanos do projeto, Gerenciamento das comunicações do projeto, Gerenciamento dos riscos do projeto, Gerenciamento das aquisições do projeto e Gerenciamento das partes interessadas do projeto.

Assim, do cruzamento das 5 categorias dos grupos de processos de gerenciamento com as áreas de conhecimento, tem-se um total de 47 processos de gerenciamento identificados no Guia PMBoK, os quais são recomendações para o gerenciamento de projetos em geral. Estes 47 processos estão ilustrados no quadro 3.

Quadro 3 - Grupos de processo de gerenciamento de projetos e mapeamento das áreas de conhecimento do guia PMBoK

(continua)

Áreas de conhecimento	Grupo de Processo: Iniciação	Grupo de Processo: Planejamento	Grupo de Processo: Execução	Grupo de Processo: Monitoramento e Controle	Grupo de Processo: Encerramento
4. Gerenciamento da integração do projeto	4.1 Desenvolver o termo de abertura do projeto	4.2 Desenvolver o plano de gerenciamento do projeto	4.3 Orientar e gerenciar o trabalho do projeto	4.4 Monitorar e controlar o trabalho do projeto 4.5 Realizar o controle integrado de mudanças	4.6 Encerrar o projeto ou fase
5. Gerenciamento do escopo do projeto		5.1 Planejar o gerenciamento do escopo 5.2 Coletar os requisitos 5.3 Definir o escopo 5.4 Criar EAP		5.5 Validar o escopo 5.6 Controlar o escopo	
6.		6.1 Planejar o		6.7 Controlar o	

Quadro 3 - Grupos de processo de gerenciamento de projetos e mapeamento das áreas de conhecimento do guia PMBoK

(continuação)

Áreas de conhecimento	Grupo de Processo: Iniciação	Grupo de Processo: Planejamento	Grupo de Processo: Execução	Grupo de Processo: Monitoramento e Controle	Grupo de Processo: Encerramento
Gerenciamento do tempo do projeto		gerenciamento do cronograma 6.2 Definir as atividades 6.3 Sequenciar as atividades 6.4 Estimar os recursos das atividades 6.5 Estimar as durações das atividades 6.6 Desenvolver o cronograma		cronograma	
7. Gerenciamento dos custos do projeto		7.1 Planejar o gerenciamento dos custos 7.2 Estimar os custos 7.3 Determinar o orçamento		7.4 Controlar os custos	
8. Gerenciamento da qualidade do projeto		8.1 Planejar o gerenciamento da qualidade	8.2 Realizar a garantia da qualidade	8.3 Controlar a qualidade	
9. Gerenciamento dos recursos humanos do projeto		9.1 Planejar o gerenciamento dos recursos humanos	9.2 Mobilizar a equipe do projeto 9.3 Desenvolver a equipe do projeto 9.4 Gerenciar a equipe do projeto		
10. Gerenciamento das comunicações		10.1 Planejar o gerenciamento das comunicações	10.2 Gerenciar as comunicações	10.3 Controlar as comunicações	

Quadro 3 - Grupos de processo de gerenciamento de projetos e mapeamento das áreas de conhecimento do guia PMBoK

(conclusão)

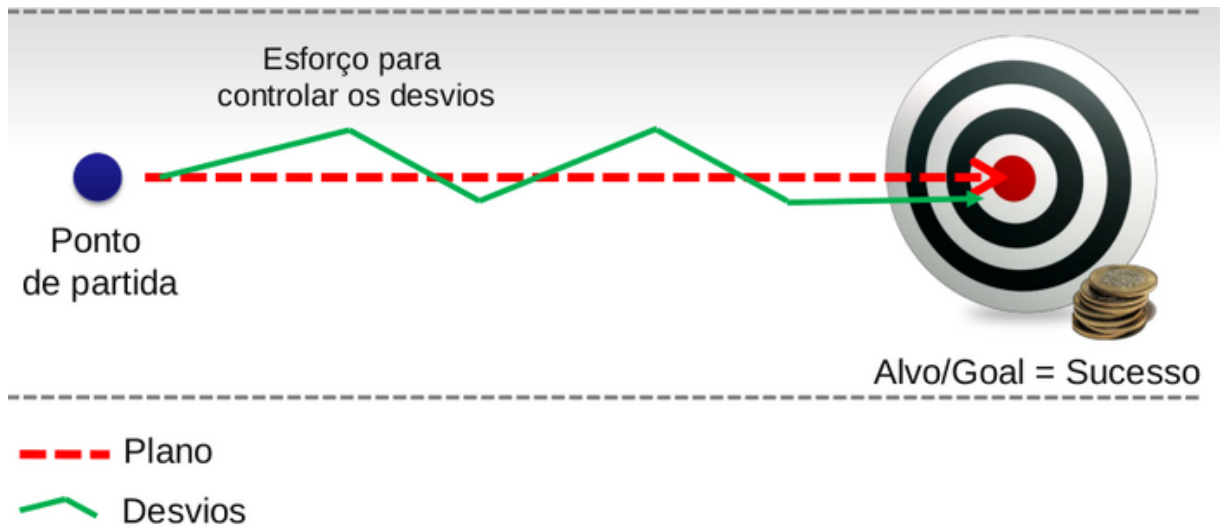
Áreas de conhecimento	Grupo de Processo: Iniciação	Grupo de Processo: Planejamento	Grupo de Processo: Execução	Grupo de Processo: Monitoramento e Controle	Grupo de Processo: Encerramento
do projeto					
11. Gerenciamento dos riscos do projeto		11.1 Planejar o gerenciamento dos riscos 11.2 Identificar os riscos 11.3 Realizar a análise qualitativa dos riscos 11.4 Realizar a análise quantitativa dos riscos 11.5 Planejar as respostas aos riscos		11.6 Controlar os riscos	
12. Gerenciamento das aquisições do projeto		12.1 Planejar o gerenciamento das aquisições	12.2 Conduzir as aquisições	12.3 Controlar as aquisições	12.4 Encerrar as aquisições
13. Gerenciamento das partes interessadas	13.1 Identificar as partes interessadas	13.2 Planejar o gerenciamento das partes interessadas	13.3 Gerenciar o engajamento das partes interessadas	13.4 Controlar o engajamento das partes interessadas	

Fonte: PMI (2013)

Nos projetos de tecnologia e desenvolvimento de software têm-se utilizado tradicionalmente a aplicação conjunta das práticas e processos do gerenciamento de projetos descritas no PMBoK. Desta forma, cria-se um plano de projeto de forma detalhada e todo o projeto é guiado ao longo deste plano, uma vez que que é estabelecido em todos seus detalhes antes de sua implementação.

Por meio do rigor de planejamento busca-se a previsibilidade e o controle sobre as entregas a serem realizadas ao fim do projeto. Assim, fixa-se o plano, e todo esforço busca controlar os possíveis desvios. A figura 3 ilustra os esforços da metodologia tradicional para se manter no plano pré-determinado.

Figura 3 - Objetivo da gestão de projeto – abordagem tradicional.



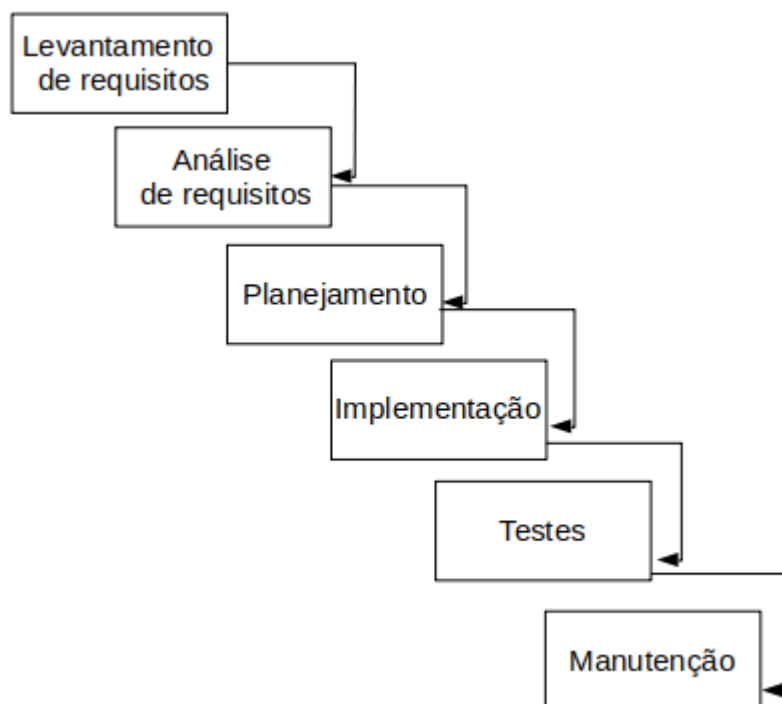
Fonte: AMARAL et al. (2011)

Segundo Pinto (2007) existem diversas metodologias para o desenvolvimento de software clássico baseado nas práticas e processos do PMBoK. O modelo cascata é o mais conhecido e antigo na indústria de software utilizado pela engenharia de software, nele os processos são muitos bem definidos com pontos de controle.

Apesar da simplicidade de gestão, o modelo cascata tem seu ciclo de vida bastante enrijecido pois ao início do projeto é desprendido um maior tempo na elaboração do plano de execução de todas as etapas necessárias ao desenvolvimento do software: o levantamento de requisito, análise de requisito, planejamento, implementação, teste e manutenção. Este modelo está explicado na figura 4.



Figura 4 - Etapas modelo cascata



Fonte: Sommerville (2011)

Neste modelo, a fase seguinte não deve ser iniciada até que a fase anterior seja concluída. Assim quando uma fase é concluída, segundo Sommerville (2011), os documentos produzidos em cada fase podem ser modificados para refletir as alterações feitas em cada um deles. As interações podem ser dispendiosas e envolver significativo retrabalho.

Ao contrário do método ágil, os métodos tradicionais, ainda segundo Sommerville (2011) devem ser usados apenas quando os requisitos são bem compreendidos e pouco provavelmente venham a ser radicalmente alterados durante o desenvolvimento do sistema.

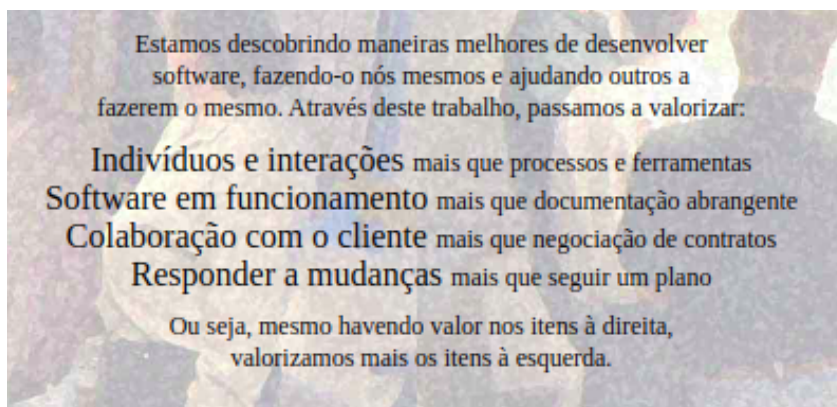
## 2.2 MÉTODOS ÁGEIS DE GERENCIAMENTO DE PROJETO

Nos dias atuais muitas empresas operam em um ambiente global e com mudanças rápidas. Diante disso, muitas necessitam de softwares para suas operações de negócio mas, devido às rápidas mudanças, é praticamente impossível obter um conjunto completo de requisitos de software estável. Atualmente, os requisitos iniciais são alterados devido à dificuldade de prever uma necessidade futura e as mudanças que ocorrem (SOMMERVILLE, 2011).

Devido às necessidades atuais de desenvolvimento rápido e de processos capazes de lidar com mudanças nos requisitos, surgiu em 2001 a abordagem de

desenvolvimento ágil de software. A teoria teve início em um movimento da comunidade internacional, que se reuniu e elaborou um documento chamado: Manifesto Ágil. Conforme já mencionado na introdução deste trabalho, este manifesto é uma declaração de valores e princípios para o desenvolvimento de um software. Na figura 5 estão descritos os valores do manifesto ágil.

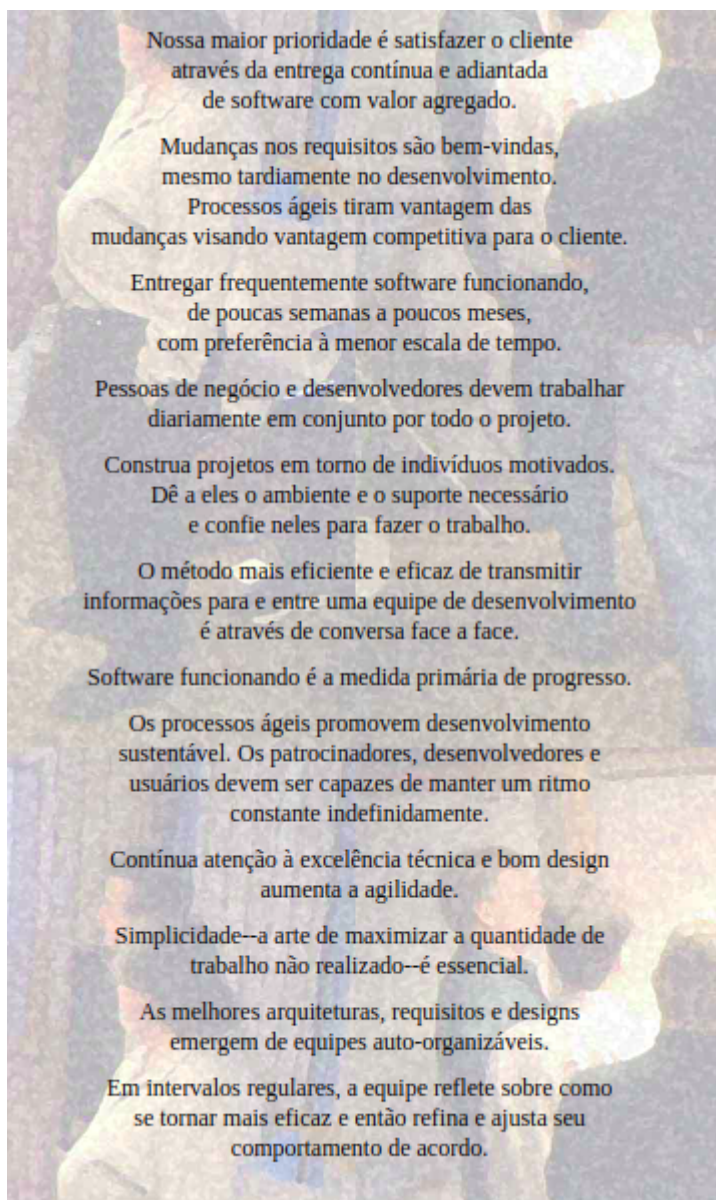
Figura 5 - Valores do manifesto ágil



Fonte: agile (2001)

Os princípios do manifesto se tornaram uma espécie de guia que orienta as ações, as escolhas de métodos e ferramentas dos times ágeis de projetos, maximizando os resultados para o sucesso de um projeto de desenvolvimento ágil de software. Tais princípios podem ser observados na figura 6.

Figura 6 - Princípios do manifesto ágil



Fonte: agile (2001)

Os métodos ágeis expõem a necessidade de um planejamento adaptativo, simples e de entrega contínua com valor agregado. A união do feedback e rápidas alterações está estritamente ligado a este método, concebidos para aceitar uma grande quantidade de alterações na sua concepção, segundo Williams e Cockburn (2003).

A aceitação de alteração do projeto durante a sua concepção, entre outras características da gestão de projetos ágeis, pode ser observada no quadro 4, onde é feito uma comparação entre o gerenciamento de projeto tradicional e o gerenciamento de projeto ágil.

Quadro 4 - Comparação entre GPA e GP "tradicional"

Gestão de Projeto Tradicional	Gestão de Projetos Ágil
As metas do projeto tem enfoque na finalização do projeto no tempo, custo e requisitos de qualidade.	As metas do projeto tem enfoque nos resultados do negócio, atingir múltiplos critérios de sucesso.
O plano do projeto tem uma coleção de atividades que são executadas como planejado para atender a restrição tripla (tempo, custo e qualidade).	O plano do projeto tem uma organização e processo para atingir as metas esperadas e os resultados do negócio.
O planejamento é realizado uma vez no início do projeto.	O planejamento é realizado no início e reavaliado sempre que necessário.
A abordagem gerencial é rígida, com foco no plano inicial.	A abordagem gerencial é flexível, variável e adaptativa.
O trabalho é previsível, mensurável, linear e simples.	O trabalho é imprevisível, não mensurável, não linear e complexo.
O projeto é controlado para, identificar desvios do plano inicial e corrigir o trabalho para seguir o plano.	O projeto é controlado para, identificar mudanças no ambiente e ajustar o plano adequadamente.
A aplicação da metodologia é genérica e igualitária em todos os projetos.	A aplicação da metodologia é adaptável ao processo dependendo do tipo de projeto.
O estilo de gestão tem um modelo que atende todos os tipos de projetos.	O modelo de gestão tem uma abordagem adaptativa. Um único modelo não atende todos os tipos de projetos.

Fonte: SHENHAR e DVIR (1996)

No final dos anos 90 até os anos mais recentes, surgiram inúmeros métodos ágeis de gerenciamento de projetos para desenvolvimento de software, como 'Crystal', 'Dynamic Systems Development', 'eXtreme Programming', 'Feature Driven Development', 'Kanban', 'Lean Software Development' e o 'Scrum'. As principais metodologias e suas características podem ser vistas no quadro 5.

Quadro 5 - Principais métodos ágeis (continua)

Metodos Ágeis	Características
Crystal	Representa um conjunto de métodos cuja idéia principal é de que o processo deve ser auto-adaptativo, não possuindo processos formais de controle do projeto, além de reuniões subjetivas e informais entre a equipe.
Dynamic Systems Development	Consiste em um método bastante iterativo, voltado a facilitar a comunicação entre o gerente de projeto, a equipe de desenvolvimento e o cliente final.
eXtreme Programmin g	Bastante focado no desenvolvedor e com muito pouca ênfase em documentação, é considerado um dos métodos mais populares, e sua combinação com o Scrum

Quadro 5 - Principais métodos ágeis (conclusão)

Metodos Ágeis	Características
(XP)	representa a maioria dos casos de aplicação de métodos ágeis em desenvolvimento de software.
Feature Driven Development	O método é considerado um dos mais leves dentre os ágeis e, quando bem aplicado, um dos mais eficientes. Representa um conjunto de práticas de gerenciamento de projetos e desenvolvimento de software, balanceado entre as filosofias tradicionais e mais radicais da agilidade.
Lean Software Development	Uma adaptação dos princípios de lean production (produção magra), o sistema de produção da Toyota para desenvolvimento de software. O método é composto por um conjunto de princípios enxutos aliado a práticas ágeis, combinados em um só ambiente
Kanban	É um método que não possui iterações, ao invés disso, desacopla o planejamento, priorização, desenvolvimento e entrega, de forma que cada uma dessas atividades possa ter sua própria cadência para melhor se ajustar à realidade e necessidade que o processo demanda. O item 2.4.2 abordará este método com mais detalhes.
Scrum	É um dos métodos mais maduros, voltado ao gerenciamento de projetos ágeis. As práticas do Scrum podem ser aplicadas a projetos de outras naturezas, além do software. O item 2.4.1 abordará este método com mais detalhes.

Fonte: O autor (2020). Adaptado de SCHUCH (2009)

Dentre todos estes métodos, destaca-se o Scrum e o Kanban, abordagens enxutas de gerenciamento de projetos de desenvolvimento de produtos desenvolvido por Ken Schwaber (1945) e Jeff Sutherland (1941).

A seguir serão apresentados os métodos Scrum e Kanban. O primeiro com o objetivo de enfatizar a diferença da abordagem tradicional em relação a ágil e, o segundo por ser o método ágil adotado neste trabalho.

### 2.2.1 Método Scrum

Segundo Schwaber e Sutherland (2017) o scrum é um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível. Ele foi inicialmente desenvolvido para gerenciar times e projetos no começo dos anos 90.

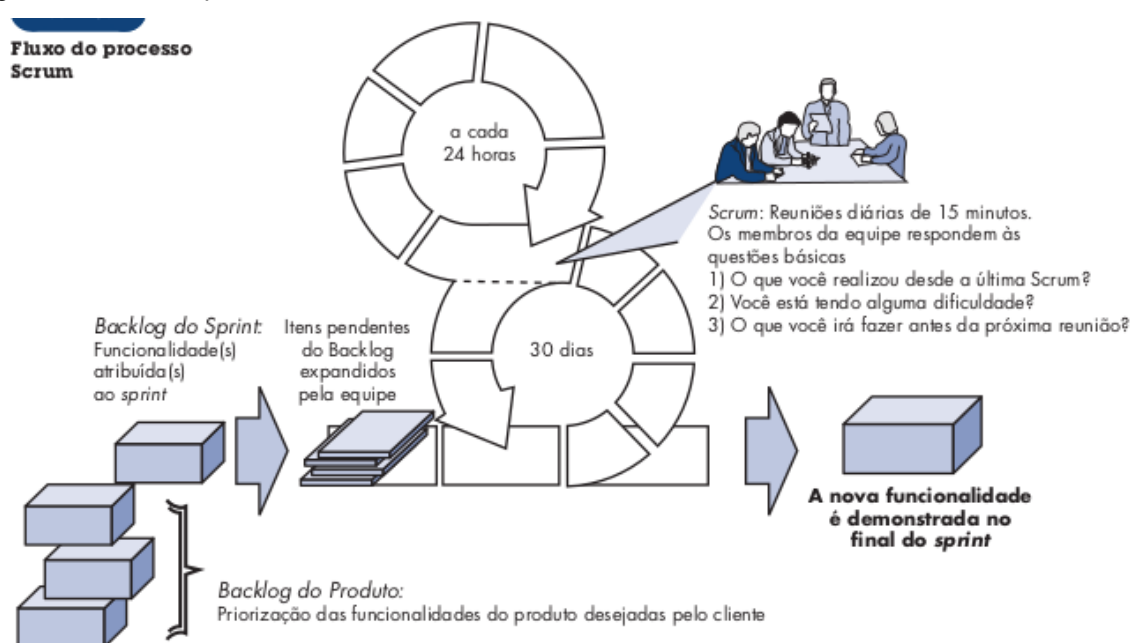
De acordo com Pressman (2011) os princípios do método são consistentes com o manifesto ágil e são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades estruturais: requisitos, análise, projeto, evolução e entrega.

As atividades no desenvolvimento do software são realizadas em "*sprints*", termo dado para um pequeno período de tempo onde um conjunto de tarefas é

executada. Segundo Pressman (2011) o número de *sprints* necessários para cada atividade metodológica varia dependendo do tamanho e da complexidade do produto em desenvolvimento.

O projeto é dividido em pequenas entregas, onde são realizados processos de planejamento, execução, testes e por fim discussão dos aprendizados. Cada entrega possui um tempo pré-determinado. Quando a etapa seguinte começa, é utilizado as informações de aprendizado e melhoria da fase anterior para constante melhoria do projeto de desenvolvimento. Este ciclo pode ser melhor compreendido na figura 7.

Figura 7 - Fluxo do processo scrum



Fonte: Pressman (2011)

As principais características da metodologia scrum são, conforme descrito na figura 9, são:

- *Sprints* são ciclos de tempos com duração fixa, geralmente de 2 a 4 semanas, geralmente correspondem ao lançamento de um pequeno acessório adicionado ao sistema;
- *Backlog* de produto, são todas as atividades que o time determinou que devem ser implementadas, correspondente a requisitos funcionais do trabalho, podendo ser novas funcionalidades ou defeitos. Processo intimamente desenvolvido com o cliente, priorizando tarefas com maior geração de valor;
- *Backlog* de *sprint*, conjunto de tarefas vindo do *backlog* de produto que serão realizadas em um *sprint*, são os requisitos dos clientes divididos em

tarefas;

- Planejamento de *Sprints*, reunião onde é feita a de avaliação do *sprint* e a seleção das tarefas a serem realizadas, são definidos os recursos e as prioridades e os riscos são identificados.
- Stand-up meeting ou daily, são reuniões diárias para analisar os progressos das tarefas e se necessário, repriorizar o trabalho ou pedir ajuda para continuar progredindo.
- *Sprint* review, uma reunião ao final de cada *sprint*, onde são compartilhadas lições aprendidas no desenvolvimento da entrega das funcionalidades definidas no *sprint backlog*.

As etapas do processo de entrega de um produto por meio do scrum podem ser divididas em: i) planejamento, que envolve toda a avaliação de qual requisito será atendido no determinado *sprint*; ii) desenvolvimento, etapa onde a equipe de desenvolvimento é isolada e trabalha para entregar o requisito do software determinado na etapa de planejamento; e, iii) encerramento, onde se tem o teste e documentação do requisito atendido.

Uma forte característica da aplicação da metodologia scrum é a definição de papéis dentro do time. Os profissionais são divididos em product owner, sendo o principal elo entre o time e o cliente. Scrum master, configurando como a figura principal do time, que lidera o andamento do projeto e os desenvolvedores caracterizando a equipe técnica.

A diversidade de projetos ágeis existentes permite que sejam combinadas metodologias para que possam trazer agilidade, melhoria contínua e geração de valor em menos tempo as empresas. De acordo com Digital.ai (2018) essas são as 10 técnicas ágeis mais empregadas nas empresas independente da metodologia ágil adotada, conforme o quadro 6.

Quadro 6 - Técnicas ágeis mais adotadas (continua)

Ranking	Técnica	Porcentagem de Adoção (%)
1	Reunião Diária	90
2	Planejamento da Sprint	88
3	Retrospectivas	85
4	Revisão da Sprint	80
5	Iterações Curtas	69
6	Planejamento de Releases	67
7	Planning Poker	65
8	Kanban	65
9	Dedicação do Product Owner	63

Quadro 6 - Técnicas ágeis mais adotadas (conclusão)

Ranking	Técnica	Porcentagem de Adoção (%)
10	Equipe Única	52

Fonte: Adaptado de Digital.ai (2018)

Conforme visto no quadro 6, uma técnica ágil utilizada pelas empresas é o Kanban, que pode empregar o papel associativo a outra metodologia como o Scrum. Esse arranjo de metodologias originou o Scrumban, um termo comum em times ágeis para designar um método formado pelas duas práticas.

Os benefícios desta associação segundo Scrum.org (2018), é promover o foco em melhorar o fluxo por meio do ciclo de feedback, aumentando a transparência e a frequência de inspeção além da adaptação tanto para o produto quanto para o processo. A seguir será apresentado a da metodologia Kanban.

### 2.2.2 Método Kanban

Kanban é uma palavra que vem da língua japonesa e que significa literalmente “cartão de sinal”. O sistema kanban teve origem no chão de fábrica da empresa japonesa Toyota na década de 40 com o objetivo de garantir seus níveis de estoques mais eficientes.

Embora esse sistema seja um conceito relativamente novo em TI, ele vêm sendo utilizados por mais de 70 anos no sistema de produção industrial. Segundo Leite et al. (2004) as principais características do sistema kanban são:

- Melhoria total e contínua dos sistemas de produção;
- Regulagem do fluxo de itens globais com controle visual a fim de executar essas funções com precisão;
- Simplificação do trabalho administrativo dando autonomia ao chão de fábrica;
- Informação transmitida de forma organizada e rápida.

O trabalho das equipes que usam a técnica, gira em torno de um quadro kanban, uma ferramenta usada para visualizar o trabalho e otimizar o fluxo do trabalho. Ao contrário da implementação do kanban em um chão de fábrica, que envolveria mudanças nos processos físicos e a adição de materiais, as únicas coisas físicas que as equipes de software precisam são um quadro e cartões, que podem até mesmo ser virtuais (RADIGAN, 2019).

Em 2004 o primeiro quadro implementado para engenharia de software aconteceu na Microsoft. Em setembro de 2006, David Anderson aplicou o sistema kanban na empresa Corbis. A análise dos resultados por David foi tão encorajadoras que o levou ao desenvolvimento da maior parte das ideias do método Kanban.



De acordo com Anderson (2011), Kanban (“K” maiúsculo) é usado para se referir à metodologia de melhoria de processo incremental e evolutiva. Assim como, “kanban” (“k” minúsculo) refere-se aos cartões sinalizadores e “kanban sistema” (“k” minúsculo) refere-se ao sistema puxado implementado com cartões sinalizadores

Segundo Neto (2017) o objetivo da metodologia é minimizar a resistência à mudança e facilitá-la, com foco no trabalho que atinge as necessidades do cliente. As características evidenciadas com as 10 práticas gerais da metodologia, são:

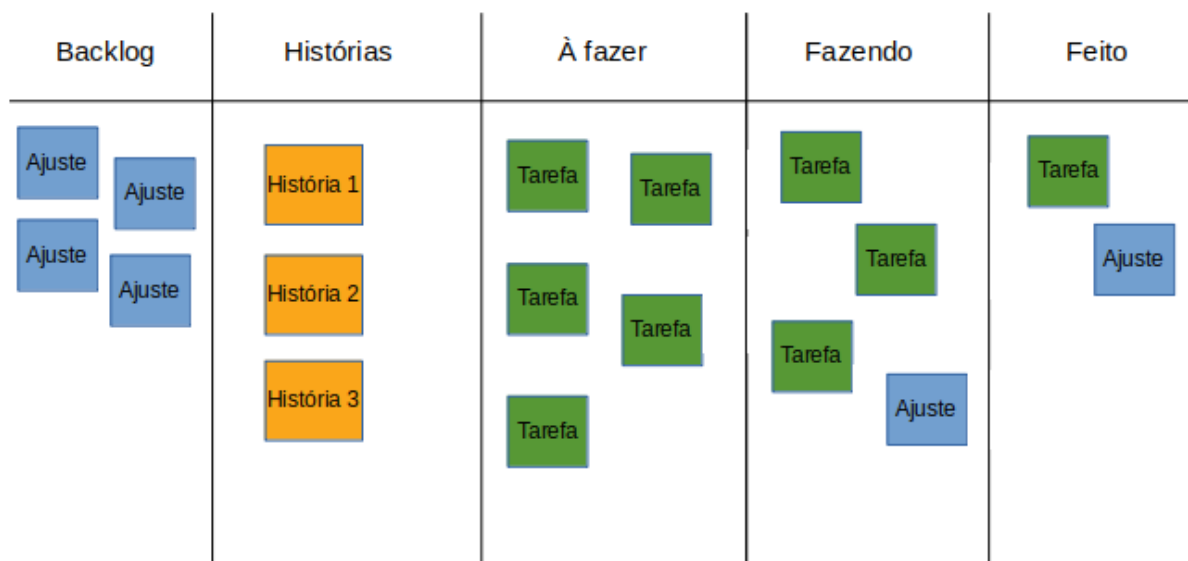
- Visualização do fluxo de trabalho;
- Limitação do trabalho em andamento;
- Definir políticas de gestão de qualidade;
- Ajustar cadência;
- Medir fluxo de trabalho;
- Definir prioridades;
- Identificar classes de serviço;
- Gerência de fluxo;
- Estabelecer acordos de nível de serviço;
- Foco na melhoria contínua

Essas práticas são aplicadas no projeto com o auxílio do quadro kanban, sistema utilizado com a finalidade de ajudar as equipes a assimilar e controlar o progresso de suas tarefas de forma visual. De acordo com Neto (2017), esse quadro exibe um conjunto de cartões que descrevem uma tarefa nova ou um débito técnico a ser resolvida. Esses cartões são organizados no quadro por colunas, que representam o estado de desenvolvimento em que uma tarefa se encontra.

Por exemplo, o quadro ilustrado na figura 8 possui 5 colunas, sendo ela:

- *Backlog*: São o escopo e os requisitos do produto que precisam ser entregues, bem como todo o entendimento necessário para se atender a esses requisitos e produzir um produto;
- *Histórias*: é uma descrição concisa de uma necessidade do usuário do produto (ou seja, de um “requisito”) sob o ponto de vista deste usuário. as frases devem seguir o seguinte padrão “Como um <tipo de usuário>, eu quero <um objetivo> para que <atenda uma necessidade>.”;
- *Á fazer*: São atividades programadas;
- *Fazendo*: São atividades que estão sendo realizadas;
- *Feito*: Tarefas concluídas.

Figura 8 - Quadro Kanban



Fonte: Adaptado de Cruz (2015)

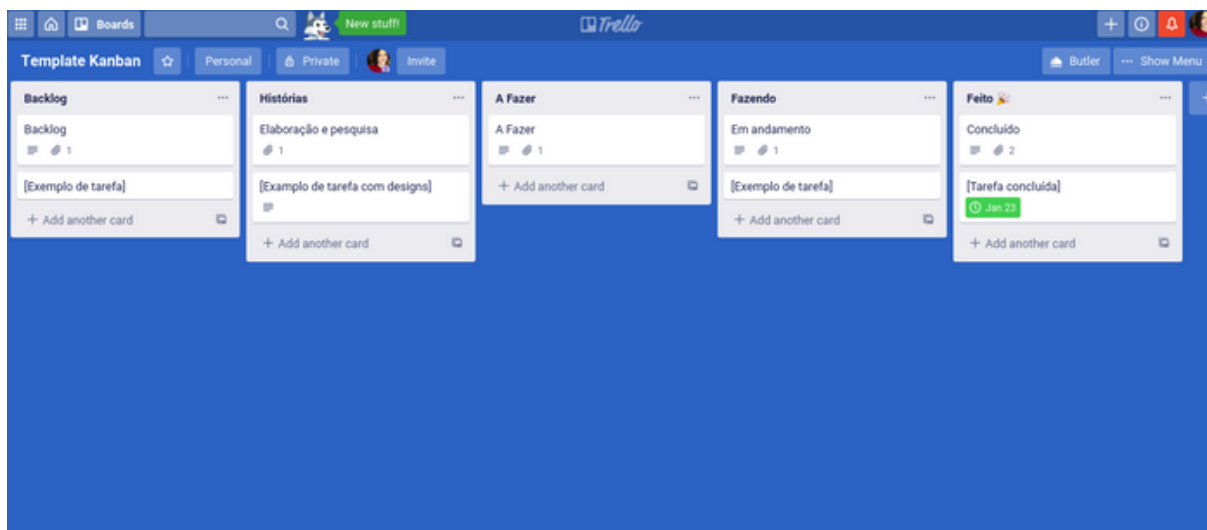
Cada cartão representa um pedaço de trabalho com valor para o cliente e contém muitas informações. As informações nos cartões devem dar poder aos indivíduos para tomarem suas próprias decisões ao puxar uma atividade descrita nos cartões.

De acordo com Genari e Ferrari (2015), a construção de um time de alto desempenho trabalhando com a metodologia abordada, deve ser baseada nas metas e nos objetivos que os mesmos deverão alcançar. O primeiro passo do gestor é fazer a escolha dos profissionais com as características e habilidades necessárias para se alcançar o objetivo.

Considerando o cenário atual em que a melhor equipe a ser formada pode estar geograficamente dispersa ou não possuir espaço físico suficiente para um quadro físico, cria-se uma demanda para ferramentas automatizadas de planejamento, monitoração e controle.

Um dessas ferramentas é o Trello, usada para simular um quadro kanban e resolver o fluxo de atividades e gestão da equipe. A Figura 9 apresenta um quadro kanban similar ao quadro kanban clássico, porém usando o Trello. Nota-se que não há perda de informações, pelo contrário, o uso do Trello permite que seja visualizado várias informações sem ocupar espaço físico.

Figura 9 - Quadro Kanban Trello



Fonte: O autor (2020)

Segundo Genari e Ferrari (2015), o Kanban em si é um processo que auxilia o desenvolvimento e a organização do projeto auxiliando na evolução de um processo já existente, visando expor os problemas de maneira mais clara. Portanto é de grande importância o quadro estar visualmente disponível para a equipe, processo auxiliado pelo uso de novas ferramentas.

Equipes excelentes utilizam o quadro para extrair dados. Um deles, essencial na análise para que um cartão seja puxado, é a data que o item entrou no sistema. Este dado permite que os membros da equipe vejam em quantos dias foi feita a tarefa em relação ao combinado de nível de serviço, uma das métricas da metodologia.

De acordo com Neto (2017) apesar do Kanban ser uma abordagem ágil, incremental, evolutiva e de fácil compreensão e manutenção, esta abordagem altera a maneira como a equipe interage com os seus parceiros. Para que essa relação entre parceiros e equipe seja a melhor possível, um desenvolvedor pode trabalhar em apenas uma tarefa por vez, além da metodologia descrever a importância de reportar métricas. Estas, estabelecidas por Anderson (2011) e Scrum.org (2018), são:

- **Work in Progress (WIP):** É o que está em execução (tarefas) naquele determinado ponto do processo, portanto não entra na conta os já posicionados na coluna "feito". A limitação de WIP é uma das propriedades centrais da metodologia, ela restringe o número de cartões nas diferentes colunas, pois estudos comprovaram que quanto maior o número de tarefas em andamento em determinada parte do processo, maior o lead time.
- **Lead time:** Tempo que uma tarefa leva para se movimentar da coluna "à

fazer" para a coluna "feito", isto é, o tempo para uma tarefa ser realizada dentro de todo o sistema;

- **Tempo de ciclo:** Tempo que uma tarefa demora para se movimentar da coluna "fazendo" a "feito", ou seja, tempo em que alguém do time está realmente comprometido e trabalhando na entrega.
- **Cumulative Flow Diagram (CFD):** Diagrama de fluxo cumulativo que mostra as quantidades de trabalho em progresso em cada estágio do sistema, ou seja em cada coluna do quadro kanban;
- **Throughput:** Número de itens de trabalho "concluídos" ou "feito" por unidade de tempo;
- **Coeficiente de variação (CoV):** Análise de quantas tarefas foram entregues durante um determinado tempo estabelecido, podendo ter uma comparação entre uma unidade de tempo e outra, através um coeficiente de variação entre elas.

Apesar de não existir nenhuma definição dos ritos praticados no Kanban, algumas cerimônias podem ser incorporadas para que se possa manter a visibilidade e planejamento do trabalho a ser realizado. Além disso, trazem medição e gerenciamento do fluxo existente, as principais cerimônias são:

- Reuniões standup, que são reuniões diárias onde se discute o fluxo do trabalho, onde o facilitador escolhido "andar pelo quadro" convencionalmente de trás para frente. Segundo Anderson (2011), o facilitador deve solicitar uma atualização do status em um ticket ou simplesmente perguntar se há alguma informação adicional que não está no quadro e pode não ser conhecida pela equipe.
- Após reunião que consiste em uma conversa com pequenos grupos para se aprofundar ou discutir algo que a equipe estava pensando
- Reuniões para reabastecimento de fila, onde servem o propósito de priorização da atividades. Feitas em intervalo regulares para que o projeto tenha uma cadência constante gerando redução de custo;
- Reuniões de Planejamento de Release ou waves onde se tem o planejamento da entrega ao final de uma cadeia de valor. Onde perguntas como: Quais itens estão prontos? Quais os riscos envolvidos? Quais são os planos de contingência? Entre outras, são respondidas.
- Triagem, termo utilizado para classificar bugs que serão corrigidos, e sua prioridade, versus bugs que não serão corrigidos e será permitido que escapem para produção quando o produto for feito o lançamento.

Apesar do Kanban e do Scrum compartilharem de muitos conceitos, possuem abordagens diferentes como pode ser visto no quadro 7.

Quadro 7 - Scrum vs. Kanban

	SCRUM	KANBAN
Cadência	Regular sprints de comprimento fixo (ou seja, 2 semanas).	Fluxo contínuo.
Metodologia de lançamento	No final de cada sprint se aprovado pelo proprietário do produto.	Entrega contínua ou a critério da equipe.
Funções	Proprietário do produto, scrum master, equipe de desenvolvimento.	Não há papéis pré-definidos para os times. Algumas equipes recebem a ajuda de um técnico do Agile.
Métricas importantes	Velocidade.	Tempo de ciclo.
Mudança de filosofia	As equipes devem se esforçar para não fazer mudanças na previsão do sprint durante o sprint. Fazer isso compromete as aprendizagens em torno da estimativa.	A mudança pode acontecer a qualquer momento.

Fonte: Radigan (2019)

Em suma, o grande foco das metodologias ágeis é permitir que o time detenha a autonomia suficiente para decidir quais tecnologia serão utilizadas, quais os ritos serão adotados, os processos que serão seguidos, qual a metodologia de desenvolvimento a ser empregada e quais as melhores maneiras de organizar a demanda de seu trabalho. As metodologias preconizam que se deve ter liberdade suficiente para buscar o objetivo do projeto a ser entregue e a sua auto organização, conceito destacado no 11º princípio do manifesto ágil, chave para construir times de alto desempenho (AGILE, 2001).

A seguir serão apresentados conceitos básicos de software e seu desenvolvimento.

### 2.3 SOFTWARE

De acordo com Silva Filho (2015), ao longo das últimas décadas software deixou de ser uma parte pequena e de custo desprezível dos sistemas para se tornar parte determinante e com alto valor no mercado. Um software é onipresente, está e estará presente em todo lugar, cunhando um conceito que surgiu com a expressão "*software everywhere*". Em suma, softwares podem ser encontrados em computadores, telefones celulares, aviões, meios de pagamento eletrônico, farmácias e tantos outros lugares e aplicativos.

Segundo Pressman (2011), um software consiste em:

- Instruções (programas de computador) que, quando executadas fornecem

características, funções e desempenhos desejados;

- Estruturas de dados que possibilitam aos programas a manipulação de informações de forma adequada;
- Informação descritiva, tanto na forma impressa como na virtual, descrevendo a operação e o uso dos programas.

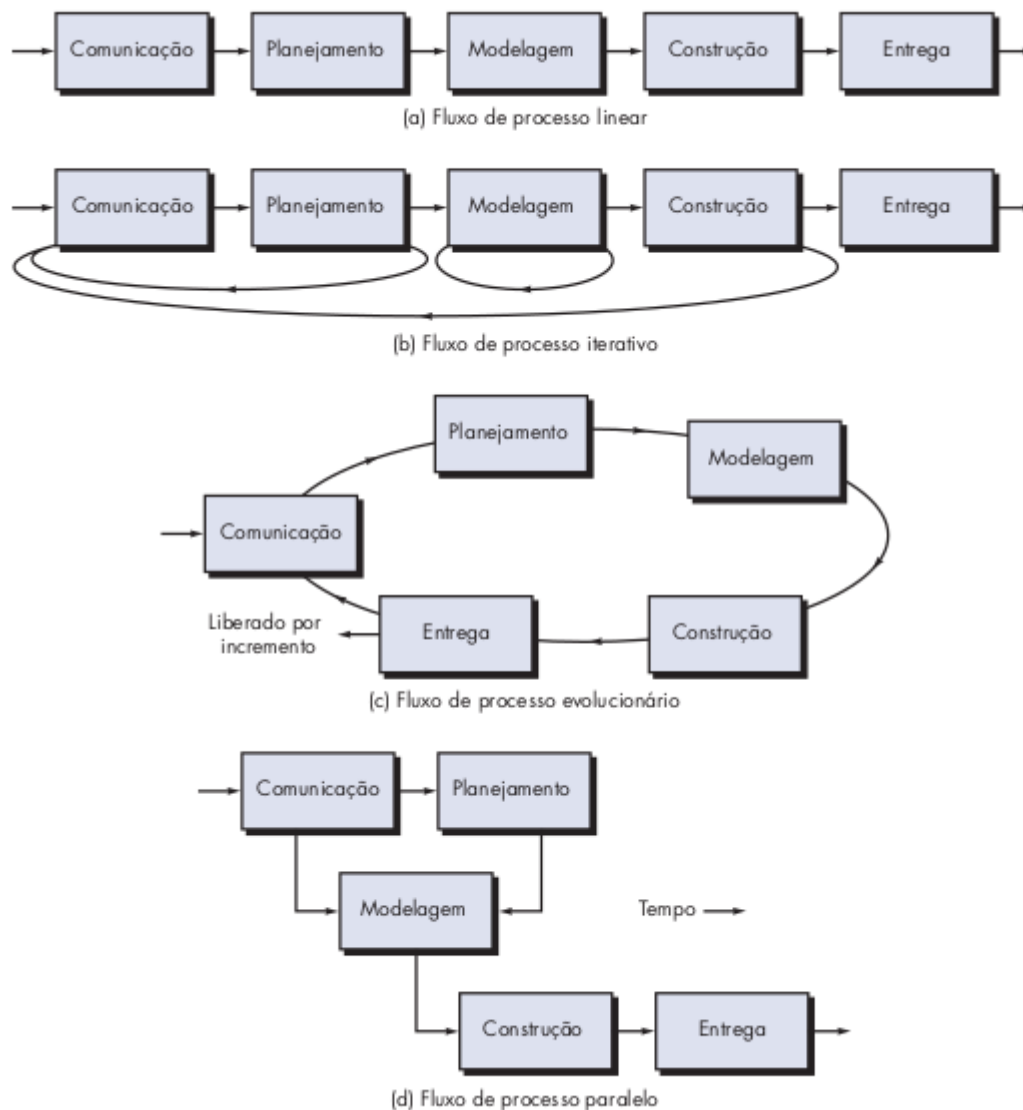
A qualidade do produto é obtida por meio de um bom projeto, pois cada defeito do software indica um erro no projeto ou no processo. Silva Filho (2015) coloca que para ter sucesso na execução do projeto, o profissional responsável precisa considerar três aspectos essenciais aos pilares da engenharia: custo, tempo e qualidade.

Sendo assim, se torna cada vez mais relevante um bom planejamento e gestão de projeto, para que o objetivo final de atender aos requisitos do usuário seja contemplado. Segundo Santos e Schmidt (2006), a palavra “usuário” não se refere apenas as pessoas, mas também pode designar outros sistemas que tenham que interagir com o software em questão.

## 2.4 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

O processo de desenvolvimento de software constitui a base para um projeto de software e auxilia a equipe na definição dos objetivos e funcionalidades a serem implementadas, nos métodos e nas técnicas a serem utilizadas, assim como, na definição dos marcos do projeto. Uma metodologia de processo genérica para um projeto em engenharia de software, estabelece cinco etapas metodológicas, podendo ter diferentes fluxos de processo de acordo com Pressman (2011), como ilustrado na figura 10.

Figura 10 - Fluxo de processos



Fonte: Pressman (2011)

As etapas gerais metodológicas comuns são: i) Comunicação e a colaborações com o cliente; ii) Planejamento do projeto do software; iii) Modelagem para melhor entender as necessidades do software; iv) Construção com a geração de código e testes; e, v) Entrega final para o cliente. Os diferentes fluxos de desenvolvimento são: i) Linear, onde se executa uma tarefa seguida da outra; ii) Iterativo, que se repete uma ou mais atividade; iii) Evolucionário, com uma execução de forma circular; e, iv) Paralelo, executando uma ou mais tarefas ao mesmo tempo.

As etapas de comunicação, planejamento e modelagem contém as atividades de levantamento de requisito, análise de requisitos e planejamento, as quais contemplam o projeto do software, propriamente dito. A etapa de construção envolve implementação e testes. A etapa final envolve a entrega para o cliente, a qual

envolve atividade de manutenção do software. A seguir, estas atividades serão detalhadas: levantamento de requisitos, análise de requisitos, planejamento, implementação, testes e manutenção.

#### **2.4.1 Levantamento de requisitos**

O levantamento de requisitos é a primeira etapa do desenvolvimento de software e também considerada a mais importante.

Segundo Pressman (2011), a intenção desta etapa é compreender os objetivos das partes interessadas com o projeto, fazer o levantamento dos requisitos (necessidades) dos clientes e definir as funções e as características do software. Nesta etapa, ocorre a transformação da visão do cliente em uma visão única entre desenvolvedores e clientes para que, em conjunto com os clientes, possam priorizar as dores a serem resolvidas do usuário final. Esta fase é conduzida principalmente, por meio de entrevistas com o cliente e pode ser considerada a etapa onde todos os desejos do cliente identificados e validados.

Diversos projetos são abandonados nesta fase por falta de levantamento e entendimento das necessidades e requisitos. Nesta fase também é comum a empresa considerar que não é possível viabilizar a continuação do projeto.

#### **2.4.2 Análise de Requisitos**

Compreende a fase na qual a equipe de projeto se reúne para estudar todos os levantamentos realizados pelos clientes e define o estado das informações coletadas para entender se existem ambiguidades, contradições, requisitos ocultos ou incompletos e resolver estes problemas.

Ainda, nesta etapa do processo de desenvolvimento, tem-se o estudo de viabilidade dos requisitos visando a definição das especificações de projeto, para que possa ser realizada a validação e a análise dos mesmos. Ao final, é elaborado um documento que contém uma lista dos requisitos especificados, de acordo com as exigências do cliente.

Como resultado desta etapa, são formulados dois modelos de documentação de requisitos. Segundo Sommerville (2011), um documento é para os usuários finais e ou clientes, os quais precisam de uma declaração de requisitos em alto nível. A outra documentação é destinada para os desenvolvedores de sistemas com a necessidade de uma especificação mais detalhada do sistema. Esta fase de desenvolvimento tem a liberdade de criar estratégias de solução sem se preocupar com a implementação, apenas focando no problema.



### 2.4.3 Planejamento

De acordo com Pressman (2011) qualquer percurso complicado pode ser simplificado caso exista um mapa. Um projeto de software é uma jornada complicada, e a atividade de planejamento cria um “mapa” que ajuda a guiar a equipe na sua jornada. É a etapa que antecede o desenvolvimento, feita para mapear as estruturas e funcionalidades identificadas e já priorizadas na análise de requisitos.

Muito importante dentro do desenvolvimento de um software esta etapa projeta a arquitetura do programa, buscando identificar qual linguagem de programação irá ser usada, a estrutura e o tipo de banco de dados, a interface, quais são os softwares que devem fazer parte da solução, entre outros.

### 2.4.4 Implementação

Nesta etapa o sistema é codificado para a linguagem definida no projeto. Sommerville (2011) coloca que o desenvolvimento de um programa para implementar o sistema decorre naturalmente dos processos de projeto de sistema. Após o projeto estar pronto para implementação, a etapa seguinte de execução diz respeito ao trabalho técnico do desenvolvedor, que é uma fase que envolve decisões pessoais do time de desenvolvimento.

### 2.4.5 Testes

Envolve diversas atividades com o objetivo de validar o produto (software) Pinto (2007). São testadas cada funcionalidade de cada módulo, buscando, levando em consideração as especificações realizadas na fase de projeto. De acordo com Sommerville (2011), os programadores fazem alguns testes do código que estão desenvolvendo, o que, muitas vezes, revela as falhas, importantes para gerar hipóteses de casos de uso dos usuários finais e como resolver.

O objetivo desta fase é mostrar que um software atende as especificações de projeto, ao mesmo tempo que, satisfaz as especificações do cliente do sistema (SOMMERVILLE, 2011).

### 2.4.6 Manutenção

A importância da etapa de manutenção é justificada, uma vez que, em muitos casos, não é possível prever o avanço tecnológico e a quantidade de dados que

geramos na nossa sociedade. Pressman (2011) aponta que os programas de computador precisam ser corrigidos, adaptados e ampliados à medida que o tempo passasse. A realização dessas atividades de “manutenção” absorve mais pessoas e recursos do que todo o esforço aplicado na criação de um novo software.

Pressman (2011), coloca que conforme aumenta a importância do software, a comunidade da área tenta desenvolver tecnologias e metodologias que tornem mais fácil, mais rápido e mais barato desenvolver e manter programas de computador de alta qualidade. Algumas dessas metodologias diz respeito ao projeto de gerenciamento da construção de um software envolvendo todas as etapas descritas anteriormente.

Em suma, considerando que existe uma crescente demanda de software que exige cumprimento de um atendimento mais ágil por parte das empresas, no processo de desenvolvimento, se faz necessário a aplicação da melhor metodologia de gerenciamento de projeto a fim de proporcionar o melhor para os clientes.

A seguir será apresentado um estudo de caso de implementação do gestão ágil de projetos, utilizando a metodologia Kanban no desenvolvimento de software em uma empresa de marketing digital.

### 3 ESTUDO DE CASO

Neste capítulo é apresentado um estudo de caso de caráter exploratório, de abordagem qualitativa para análise e interpretação de dados, com o objetivo de definir o problema a ser abordado. Assim, de acordo com Gil (2002) este estudo visa proporcionar maior familiaridade com o problema, a torná-lo explícito ou a construir hipóteses. Considera os mais variados aspectos relativos ao fato estudado incluindo entrevistas com pessoas que tiveram experiências práticas com o problema pesquisado.

A abordagem qualitativa possui o ambiente natural como fonte direta para coleta de dados e o pesquisador é o instrumento-chave, é descritiva, o processo e seu significado são os focos principais de abordagem (SILVA; MENEZES, 2005).

Do ponto de vista técnico o trabalho faz uso de pesquisa bibliográfica que tem como característica o uso de material já publicado para fundamentar a pesquisa e trazer novos conhecimentos ao pesquisador (SILVA; MENEZES, 2005).

Assim, em relação a natureza do estudo caracteriza-se em forma de estudo de caso, pois envolve a aplicação de uma metodologia em um projeto de uma empresa, ou seja, o uso da teoria num projeto real.

Assim, inicialmente, buscar-se-á caracterizar a empresa e identificar o problema a ser tratado. E, na sequência, é mostrada a proposta apresentada para resolução do problema com base no referencial teórico apresentado no Capítulo anterior.

#### 3.1 CARACTERIZAÇÃO DA EMPRESA

A empresa, surgiu na dificuldade de outras empresas em anunciar um serviço ou produto e como fazer essa comunicação com o cliente para concretizar a venda. Ela tem o foco em prestar serviço a empresas de todos os portes a elaborar e acompanhar o sucesso do planejamento com uma boa estratégia de Marketing Digital.

Um dos produtos é voltado para empresas que queiram fazer a automatização do marketing digital. Este programa foi criado para ser intuitivo, prático, abrangente e engloba todas as necessidades existentes na realização do processo de marketing. Algumas delas são:

- Planos de Sucesso: criação de ações com tarefas, prazos e responsáveis;
- Otimização do Google: monitoramento e identificação de otimização de página e painel de palavras-chaves;
- Postagens em Mídias Sociais;

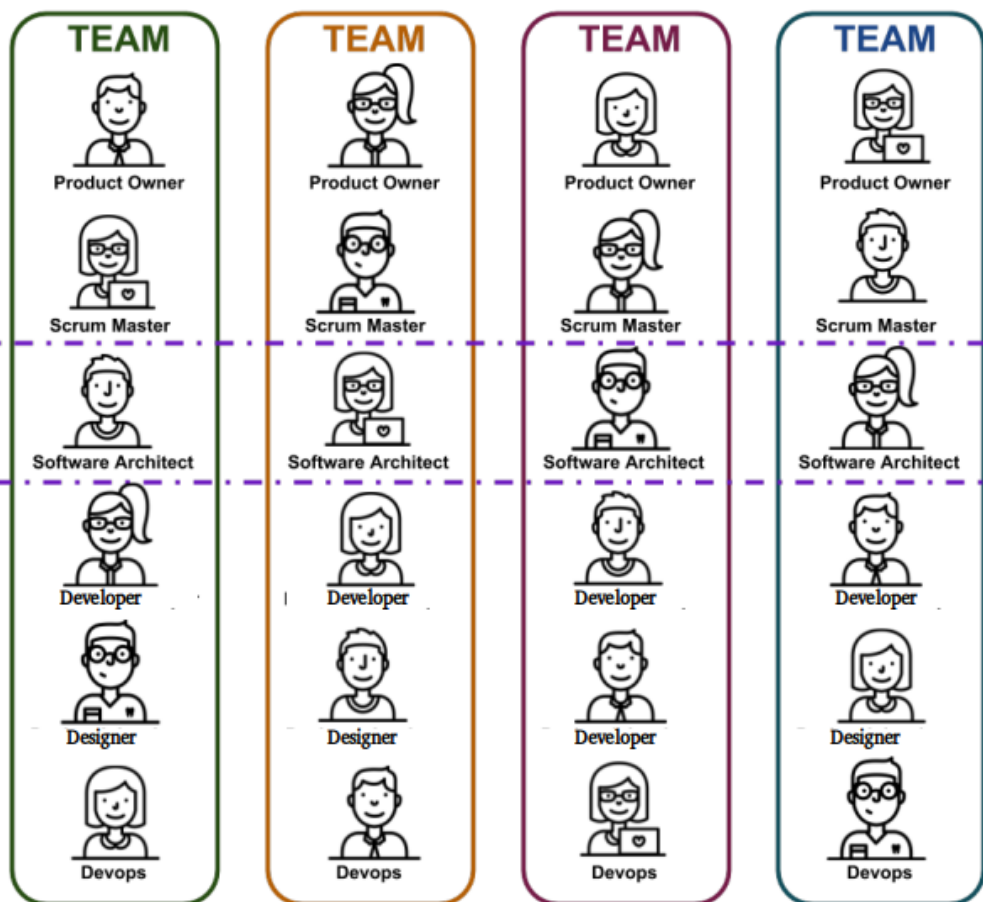
- Criação de páginas de conversão para novos clientes, notificações e formulários de forma simples;
- Gerenciamento e histórico completo de cada contato;
- Criação de campanhas de email;
- Análise de estatística dos resultados.

Como acima descrito, a empresa tem um software classificados como SaaS, sigla do inglês que significa: *software as a service*. Para que se possa oferecer um bom produto no mercado como este é necessário:

- Uma infraestrutura consistente;
- Atender os problemas dos clientes;
- Um time competente capaz de desenvolvê-lo.

São dois times os principais responsáveis por desenvolver e assegurar a qualidade técnica do produto no mercado. A equipe de engenharia e a equipe de produto, compostas por mais de 200 pessoas, as quais podem ser estruturadas em vários times, cada qual com um gerente de produto, um scrum master, um ou mais engenheiros de software, além de um ou mais designer. A organização estrutural pode ser representada conforme a figura 11.

Figura 11 - Estrutura dos times de produto e engenharia na empresa



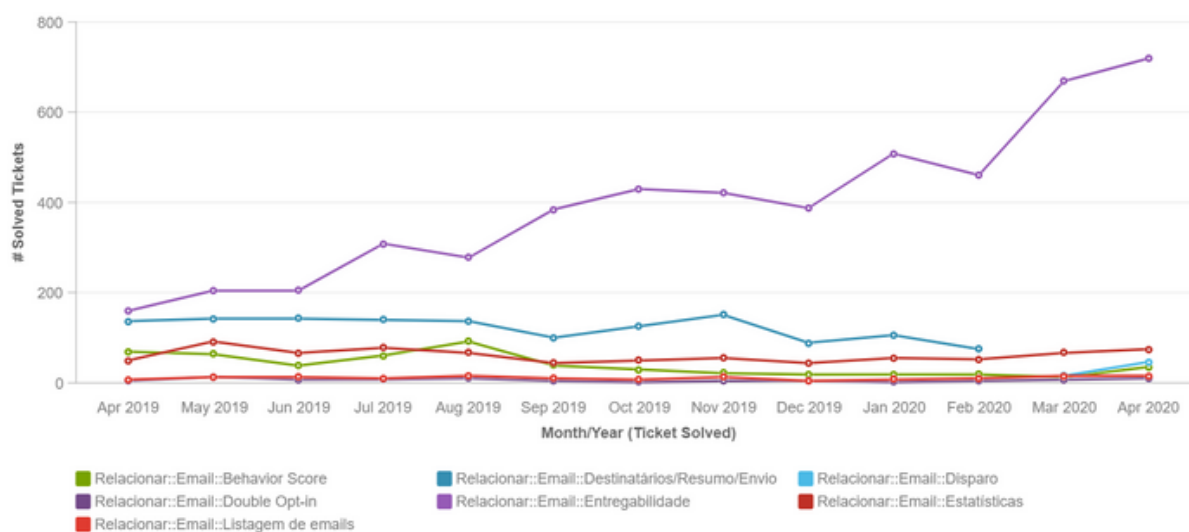
Fonte: O autor (2020)

Este trabalho trata do projeto de desenvolvimento de uma funcionalidade interna para o uso da empresa. Esta funcionalidade, em um futuro, pode vir a ser incorporada ao software principal da empresa. O desenvolvimento utiliza a metodologia ágil kanban envolvendo a área de produto e engenharia, estruturada conforme o modelo descrito na figura.

### 3.2 O CASO DE ESTUDO

Com o constante aumento do uso da criação de campanhas de emails, já citado acima como um dos processos englobados pelo produto oferecido pela empresa, todos os problemas associados a "funcionalidade de envio de email" começam a aumentar de número, conforme o mostra o gráfico da figura 12, e representado pelo nome de "tickets abertos na empresa".

Figura 12 - Tickets de email



Fonte: O autor (2020)

Na figura 12, claramente, se nota que a maior parte dos problemas relatados por nossos clientes estão associados a entregabilidade, ou seja, os emails disparados pelo cliente da ferramenta, não chegam até seus clientes (usuários, compradores, por exemplo). Por conta da importância estratégica desta funcionalidade, este tema está englobado dentro da prioridade de número dois de toda a área de produto.

Pensando em melhorar a entregabilidade dos clientes, o caso de estudo deste trabalho gira em torno de dois elementos macros:

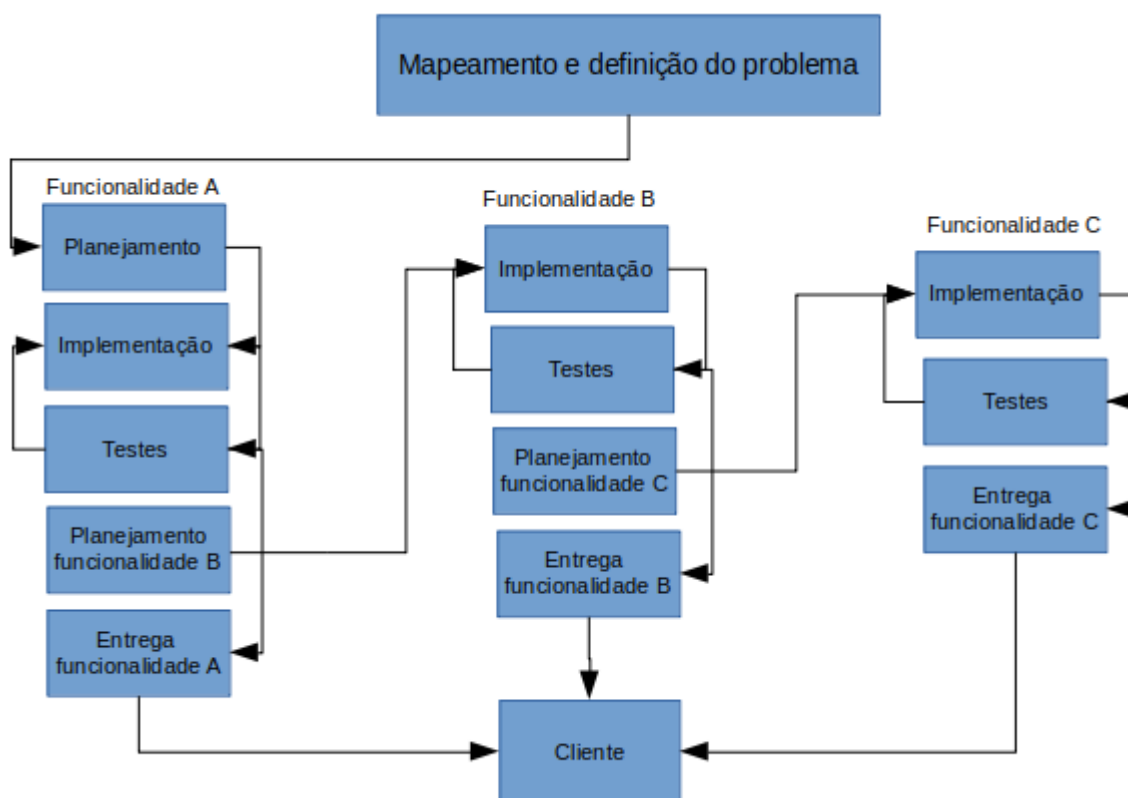
1. A reputação associada ao disparo de envio de email do cliente;
2. A reputação associada ao endereço de protocolo na internet utilizado para o disparo de email.

O segundo elemento macro envolve os endereços de protocolo na internet, que se fazem necessários para que um email seja enviado ao cliente desejado. Esse endereço funciona de forma semelhante ao CPF de uma pessoa física, permitindo que conexões e dispositivos sejam identificados a partir de uma sequência numérica. O protocolo que na área de engenharia de software se chama IP (Internet Protocol) que é um número identificador dado ao seu computador, ou roteador, ao conectar-se à rede. É através desse número que seu computador pode enviar e receber dados na internet. O IP é definido pelo seu provedor de Internet.

Historicamente estes dois problemas macros são bem distribuídos, tendo quantidades representativas de clientes sofrendo por ambos. Porém, a perda de reputação nos IPs dos nossos clientes é normalmente um problema onde a empresa atua mais ativamente, pois a alocação desses IPs em cada conta é de responsabilidade da empresa.

Para o desenvolvimento do projeto de software com o objetivo da automatização do processo de consulta de IPs, foram definidas as seguintes etapas correspondentes ao desenvolvimento do software. Etapas representadas na Figura 13.

Figura 13 - Processo genérico de desenvolvimento das funcionalidades de um software



Fonte: O autor (2020)

Conforme demonstrado na figura 13, a partir do mapeamento do problema deve ocorrer a implementação das funcionalidades A, B e C. A implementação das funcionalidades são pacotes de funções do software que tem o objetivo de entregar valor para o cliente a cada fim de ciclo. Embora a implementação das funcionalidades A, B e C estejam representadas de forma sequencial é importante ressaltar que a implementação das mesmas pode ocorrer em paralelo, a depender do problema a ser tratado e da definição das ondas de entrega.

A “funcionalidade A” envolve as etapas de planejamento, implementação, testes, planejamento da implementação da “funcionalidade B” e entrega da “funcionalidade A”, propriamente dita. Consiste em tornar a solução acessível tanto para pessoas quanto para aplicações externas. Neste passo ainda não é necessário que se tenha dados corretos da busca de contas e seus IPs. O principal objetivo de validação é que a aplicação proposta esteja disponível para ser acessada quando

necessário, de forma a mitigar o problema da interferência humana no momento de procurar a informação.

A implementação das funcionalidades B e C seguem um fluxo semelhante. Porém, na funcionalidade B o objetivo é prover, por meio dos acessos implementados na entrega 1, informações relativas às nossas contas e IPs. Nesta entrega ainda não serão tratadas as questões de inconsistência dos dados e nem questões de quão atualizados os dados estão. Com essa entrega, pretendemos validar com o cliente a forma como os dados são apresentados para mitigar o problema do difícil uso da ferramenta técnica.

Já na funcionalidade C, a entrega garante que as informações fornecidas na entrega 2 estejam atualizadas em relação ao que consta na ferramenta técnica. Ao final desta entrega, os problemas de difícil uso da ferramenta técnica por outras áreas da empresa, além dos engenheiros de software, devem ser completamente resolvidos, de forma que não se faça mais necessário o uso da antiga ferramenta de consulta pelo cliente final.

Como já definido no referencial teórico deste trabalho, aplicar uma metodologia ágil é entregar continuamente, sempre de forma incremental e com frequentes valor para o cliente. Mais ainda, aplicar a metodologia Kanban ágil é entregar um fluxo em movimento de valor para o cliente por meio de um sistema de desenvolvimento constante de produtos, ou seja, inicialmente, a funcionalidade A, depois a funcionalidade B e, por último, a funcionalidade C.

Para o desenvolvimento do projeto foi adotado o método ágil de gestão de projeto, uma vez que se encaixa no perfil do ambiente de projeto da empresa. Para o projeto foi formada uma equipe de 5 membros das áreas de engenharia e produto, ou seja, equipe interdisciplinar realizando pesquisa e desenvolvimento de software

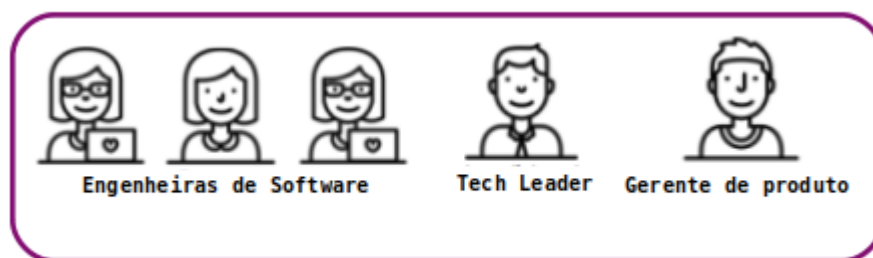
Este time foi responsável por executar o projeto, o mapeamento do problema proposto e a proposta de utilização de uma metodologia ágil com abordagem nas práticas do Kanban. O time analisou as métricas propostas na aplicação da metodologia ágil, utilizada no dia a dia para gerenciamento dos projetos.

Todos os métodos que foram utilizados foram respaldados pelas teorias de gerenciamento ágil de projetos apresentadas no capítulo 2.

O time estudado é composto por 5 pessoas, sendo elas 3 engenheiras de software, um *tech leader* e um gerente de produto, conforme a figura 14.



Figura 14 - Estrutura do time estudado



Fonte: O autor (2020)

O papel das engenheiras de software dentro do projeto em questão são desenhar, desenvolver, testar as novas funcionalidades e realizar melhorias nas funcionalidades existentes. Assim como, monitorar a operação do produto garantindo confiabilidade e disponibilidade e em caso de falha garantir uma rápida recuperação.

Já as atribuições do *tech leader* tem um perfil maior de gerência, liderando o desenvolvimento de software do time utilizando a metodologia Kanban, aplicando os processos e melhores práticas ágeis garantindo a melhor execução. Além disso, é responsável pela definição da implementação técnica da arquitetura de software dos produtos e serviços em conjunto com seu time, garantindo o aprimoramento dos padrões de código, revisão de bugs, problemas de performance e custos dos seus serviços, bem como no apoio no aprofundamento de resolução de problemas complexos dos seus times. É também o *tech leader* que faz a conexão com o a área de Produto e o time de desenvolvimento para garantir a entrega e a alta qualidade dos produtos e serviços.

Por fim, o papel do gerente de produto é identificar e avaliar oportunidades para o produto de fontes como o cliente e feedback interno de diferente áreas da sua própria empresa, além de planejar e direcionar para fazer o produto correto, representar o produto internamente e trabalhar alinhado com as diversas área.

A seguir, com base na revisão da literatura e nas informações levantadas, é apresentado um modelo de gerenciamento ágil de projetos utilizando a metodologia Kanban.

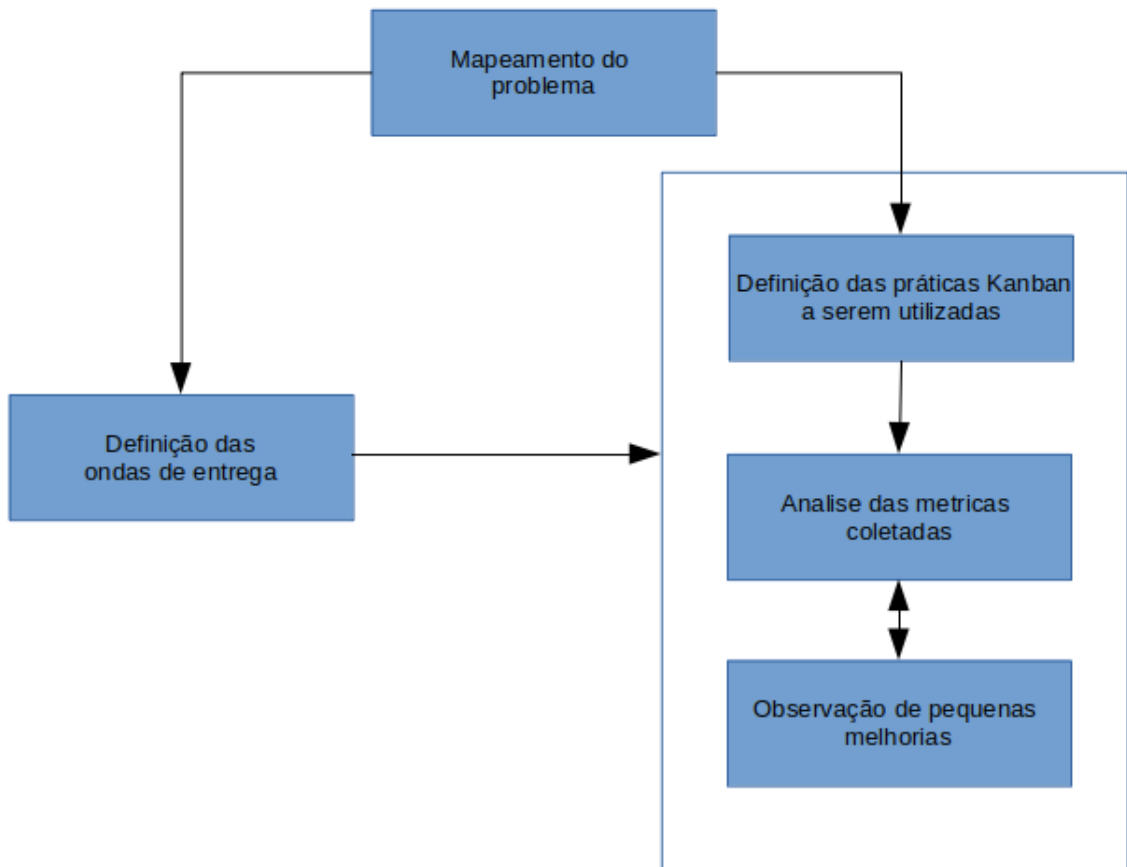
#### **4 MODELO DE GERENCIAMENTO ÁGIL DE PROJETO BASEADO NO KANBAN**

Este capítulo apresenta a proposta de modelo de gerenciamento ágil de projeto com abordagem nas práticas do Kanban, visando o desenvolvimento de software.

É importante salientar que a aplicação do modelo proposto ocorreu de forma concomitante a resolução do problema de “entregabilidade do e-mail”. Assim, ao longo de cada etapa do modelo proposto é apresentada a aplicação na empresa.

O modelo está apresentado na figura 15. No primeiro passo é realizado o mapeamento do problema a ser resolvido pela empresa, seguido concomitante da definição das ondas de entrega que se inicia quando o mapeamento do problema é feito porém pode ser um processo contínuo ao longo do desenvolvimento do software e a definição das práticas kanban a serem utilizadas segundo as necessidades do time. Conforme o decorrer do projeto, são coletadas métricas e analisadas podendo assim sugerir pequenas melhorias ao longo do projeto. A partir das melhorias implementadas, novas métricas são coletadas, sendo este um processo iterativo de melhoria contínua.

Figura 15 - Modelo de gerenciamento ágil de projetos baseados no Kanban.



Fonte: O autor (2020)

Todos os métodos descritos estão respaldados pelos referenciais teóricos de gerenciamento ágil de projetos, já apresentados no capítulo 2.

#### 4.1 MAPEAMENTO DO PROBLEMA

Os principais objetivos do mapeamento do problema são:

- Entender e identificar as insatisfações internas de quem faz parte da execução e as insatisfações externas, ou seja, de onde vem a demanda e quem recebe a demanda;
- Observar o atual cenário de tudo que está em execução, como é solucionado o problema no atual momento.
- Definir com o máximo de precisão possível a solução final do projeto;

No caso estudado, o problema em questão envolve a entregabilidade dos e-mails, assim como a reputação associada ao endereço de protocolo na internet utilizado para o disparo de email, tema no qual a empresa atua mais ativamente e

possui maior controle.com a entrada do cliente na dinâmica inicial de definição e detalhamento de atividades;

Para entender as insatisfações e observar o cenário atual do problema a ser abordado para desenvolvimento, pode ser realizado questionários, observações ou até entrevista com a entrada do cliente na dinâmica inicial de definição e detalhamento de atividades. Foi realizado na empresa, uma entrevista com dois clientes da engenharia que atendem ao ticket já mencionado. Ambos são engenheiros de software da empresa e relataram que para se consultar a reputação do IP, primeiro é necessário saber o número deste IP.

Atualmente na empresa existe um problema ao fazer essa consulta do número do IP correspondente a cada conta de cliente pois, segundo os entrevistados:

"O time responsável pelo sucesso do cliente, chamado de Heroes, precisa acessar uma ferramenta utilizada normalmente apenas por engenheiros de software, por se tratar de um recurso técnico, caso queiram procurar o número do IP. Porém, normalmente eles não possuem conhecimento para a consulta"

e ainda segundo os entrevistados essa falta de informação devido a uma consulta de difícil acesso:

"Tira a visibilidade de quão bom ou ruim estão os IPs e Contas, pois não conseguem consultar qual o número do IP que está atribuído a determinada conta e portanto não consegue, por meio deste número consultar a sua reputação"

Hoje, o processo de consulta é realizado de forma manual em arquivos confusos, de difícil visibilidade e técnicos como descrito na entrevista, podendo acarretar confusões feitas por interferência do trabalho humano e muitas dúvidas por se tratar de um arquivo complexo.

Diante deste cenário apresentado, do problema exposto, da rotina atual e do entendimento das insatisfações, foi proposto a construção de uma aplicação, ou seja, um projeto de software que irá automatizar a consulta das contas e seus IPs utilizados.

Assim, com o novo software, qualquer funcionário da empresa poderá ter visibilidade das informações de forma simples e rápida, podendo encontrar de forma mais ágil a resposta de em qual problema agir para melhorar a experiência do cliente na ferramenta.

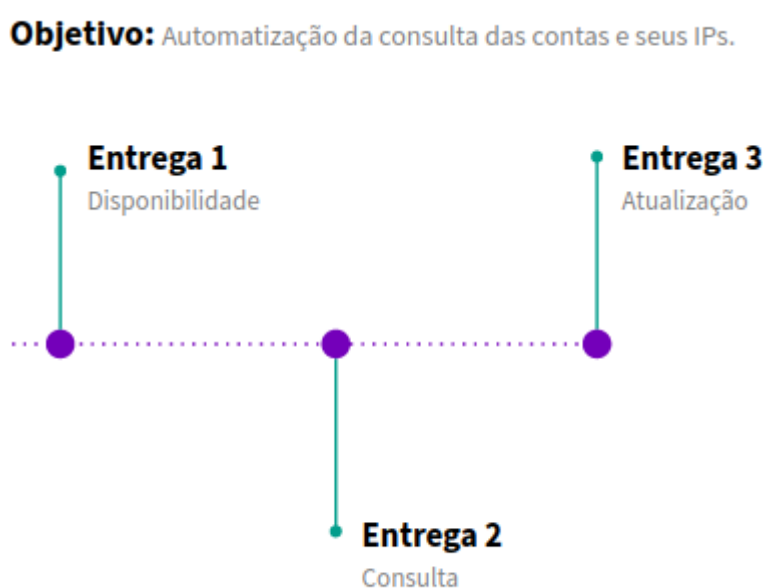
## 4.2 DEFINIÇÃO DAS ONDAS DE ENTREGAS

Para endereçar os problemas levantados de forma em que a metodologia ágil possa ajudar o time em questão a potencializar sua entrega e limitar o trabalho em andamento, foi realizada a quebra do problema em pequenas entregas com partes da solução que endereçam partes específicas da necessidade do cliente.

O principal foco na definição das ondas é sempre ao final delas entregar de alguma forma um valor para o cliente. As equipes que trabalham com a metodologia ágil focam em concluir o mínimo produto viável, para que o cliente já consiga conquistar o seu objetivo de alguma forma, e depois vão aprimorando e fazendo versões melhores.

O problema da empresa em específico foi dividido em três partes conforme a figura 16, porém a divisão de quantas ondas de entrega são necessárias é particular de cada projeto visando sempre ao final de cada etapa, o cliente.

Figura 16 - Divisão das entregas



Fonte: O autor (2020)

A quebra em pequenas entregas, além de ser uma proposta de todas as metodologias ágeis, na metodologia Kanban a divisão em pequenas entregas surge para que a resistência à mudança seja minimizada e caso ocorra seja fácil a execução.

#### 4.3 DEFINIÇÃO DAS PRÁTICAS KANBAN A SEREM UTILIZADAS

O método Kanban se caracteriza por não possuir práticas pré determinadas, sendo assim é de escolha do time adotar o que for conveniente e necessário para se ter agilidade no projeto. A equipe estudada, se utilizou do quadro kanban e de algumas cerimônias ágeis que serão descritas a seguir.

### 4.3.1 Modelagem e implementação do quadro kanban

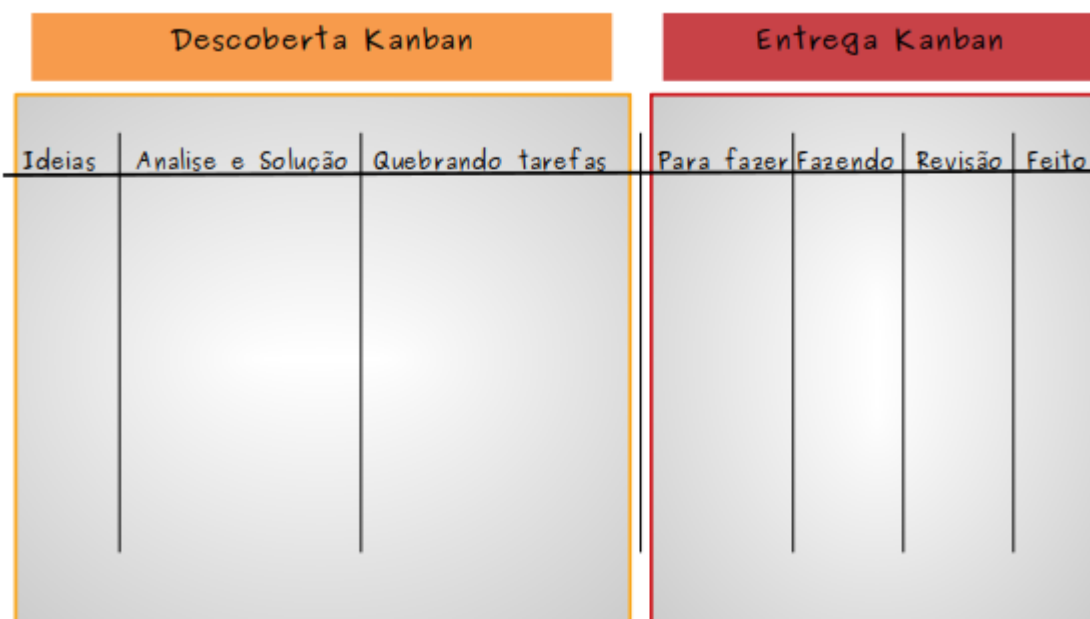
A fim de todos os participantes do projeto possam ter acesso visual de modo rápido e fácil a todas as informações para trazer mais transparência, produtividade e agilidade aos processos do projeto foi utilizado um quadro kanban visual, virtual e online, contendo os elementos mais importantes do projeto. Além das vantagens em relação a comunicação, essa ferramenta possibilita o gerenciamento da execução das atividades, gerando uma visão geral do projeto para toda a equipe, contribuindo para que todos compartilhem de uma mesma visão em relação ao projeto.

O objetivo desse quadro é que o fluxo de trabalho fique visível e que o desenvolvimento do projeto seja atualizado sempre que necessário, permitindo que todos os membros da equipe do projeto possam, quando quiserem, verificar o status do projeto, verificar as datas de entrega, as etapas realizadas e pendentes e os produtos de cada etapa.

Na figura 17 pode ser visto o quadro kanban utilizado para auxiliar o fluxo de trabalho necessário para desenvolvimento do projeto, ele foi dividido em duas partes a primeira chamada de descoberta Kanban e a segunda entrega Kanban. Na descoberta Kanban foram definidas as seguintes colunas i) Ideias: é onde os clientes e colegas de equipe colocam ideias para projetos que a equipe pode escolher quando estiver pronta para executar; ii) Análise e solução: coluna onde tarefas com uma complexidade maior são colocadas para que se possa identificar que está acontecendo um estudo antes da execução; e, iii) Quebrando tarefas é a coluna onde um card com a wave selecionada para ser executada será dividida em tarefas menores.

Já na outra parte do quadro, chamada de entrega Kanban, é onde as tarefas são saem do teórico e realmente passam a ser executadas, os processos envolvidos são i) Para fazer: são as próximas tarefas que devem ser feitas; ii) Fazendo: são as tarefas que estão em execução no momento; iii) Revisão: é a coluna que se encontram as tarefas que devem ser revisadas por outros membros da equipe para confirmarem que ela está de acordo com a qualidade esperada; e, iv) Feito: é a coluna onde são movidos os cards ou tarefas que foram concluídos.

Figura 17 - Quadro Kanban utilizado no time estudado



Fonte: O autor (2020)

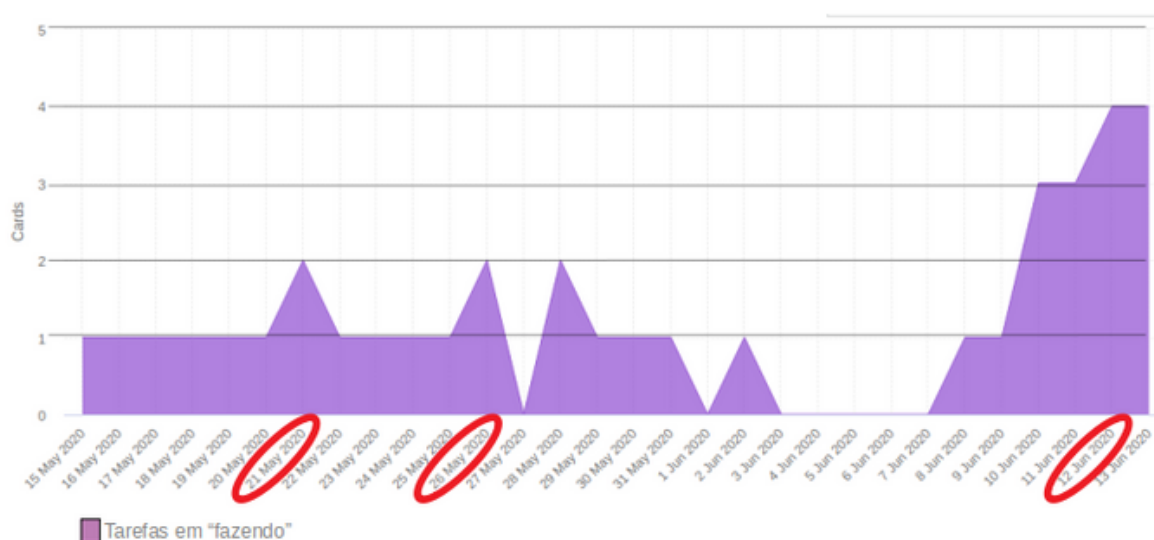
O quadro kanban é particular de cada equipe e deve ser formulado somente após o entendimento melhor do cenário, de início deve ser simples e representar a realidade atual da equipe mas conforme os conhecimentos da equipe e do projeto forem se consolidando o quadro irá mudar até ficar algo parecido com o exposto na figura 16.

#### 4.3.1.1 Limitação de quantidade de itens em progresso

Assim como já visto anteriormente no capítulo de fundamentação teórica a limitação de *WIP* (*Working in Progress*) é uma das propriedades centrais da metodologia, ela restringe o número de cartões nas diferentes colunas, pois estudos comprovaram que quanto maior o número de tarefas em andamento em determinada parte do processo, maior o tempo para aquela tarefa ser entregue.

A *WIP* pode ser determinado em um time por meio da observação do fluxo de trabalho da equipe e suas métricas. Na figura 18 é exposta uma captura do gráfico do time estudado em que é possível ver a quantidade de tarefas na coluna "fazendo" do quadro kanban ao longo do tempo.

Figura 18 - Quantidade de itens na coluna "fazendo" do quadro kanban



Fonte: O autor (2020)

É possível observar que o máximo de tarefas da equipe sempre teve um padrão ao longo do tempo, como em 21 de maio de 2020, 26 de maio de 2020 e outras datas, o limite máximo observado era de 2 tarefas em execução ao mesmo tempo. O número condiz com a realidade da equipe por ser composta de apenas 3 engenheiras de software responsáveis por desenvolver o projeto.

Porém, a partir da data de 9 de junho, houve um aumento significativo na quantidade de tarefas em andamento, chegando ao pico no dia 12 de junho. O gráfico mostra que algo de diferente aconteceu na equipe e deve ser investigado, uma vez que, na data citada como pico tem-se 4 tarefas em andamento para 3 profissionais trabalhando. A partir desta constatação, pode ser adotado a limitação de WIP. Esta é uma medida que o *tech leader* define a limitação de tarefas na coluna "fazendo", com 2 cartões, prevenindo o time de repetir o cenário do dia 12 junho.

Esta limitação de WIP no contexto de quadros Kanban, funciona como um porteiro que faz com que um novo trabalho inicie, somente após a conclusão de outro. Isso impede a acumulação de trabalho em andamento, que poderia prejudicar o fluxo de processos. A aplicação de limites de WIP no quadro Kanban ajuda a revelar os bloqueadores de processos de trabalho e previne que os membros do time mudem do contexto regular entre tarefas. Esses passos têm um impacto positivo na eficiência e traz melhoria na produtividade do time, este processo pode ser aplicado em qualquer coluna do quadro kanban, que se mostre necessário.



### 4.3.2 Implementação das cerimônias Kanban

O time responsável no projeto deve determinar as cerimônias convenientes para a equipe, assim como mencionado no capítulo 2.4.2. O Kanban não possui ritos pré determinados. No modelo desenvolvido, propõe-se adotar cerimônias com o objetivo de melhorar a gestão do projeto. As cerimônias incorporadas foram:

- **Reuniões diárias:** Semanalmente, todos os dias, foram realizadas reuniões de curta duração (até 30 minutos) com os membros do projeto. As reuniões tinham o objetivo de discutir informações gerais pertinentes ao projeto, como por exemplo andamento de atividades, status das etapas, a fim de que todos os membros tivessem conhecimento de tudo que estava acontecendo no projeto.
- **Reuniões de review:** A cada 15 dias, onde foram analisadas as métricas do projeto e identificados possíveis pontos de melhoria para equipe.
- **Reuniões de Entrega:** É feita a cada final de uma pequena entrega para se estrutura um documento com a explicação do que foi feito e como o cliente poderia utilizar a funcionalidade. Reunião também utilizada para comemorar a entrega a fim de engajar cada vez mais o time e o manter unido em prol do mesmo objetivo.
- **Reuniões de Planejamento:** Encontro de discussão da próxima entrega e o valor dela agregado para o cliente, conseqüentemente se preciso era feito uma re-priorização do trabalho. Usada também para planejar a parte técnica da próxima entrega assim como identificar o conhecimento técnico necessário para concluir o trabalho.

## 4.4 ANÁLISE DAS MÉTRICAS COLETADAS

No modelo é proposto um conjunto de indicadores utilizados para geração de métricas no kanban. Este modelo de gerenciamento por métricas embasa a teoria da metodologia Kanban já apresentada no item 2.4.2.

### 4.4.1 Tempo de ciclo

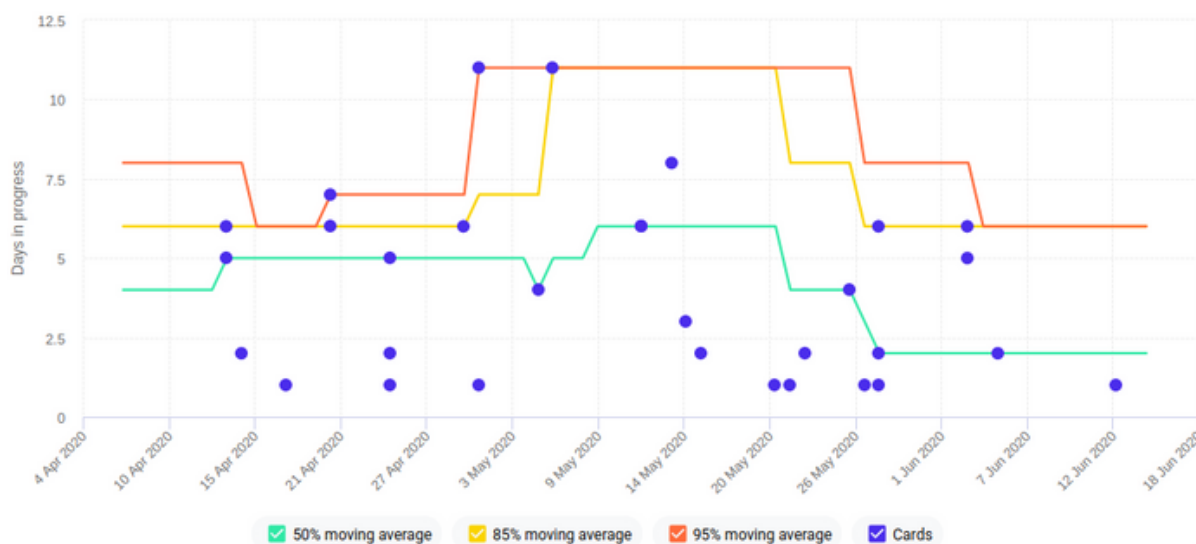
A equipe utilizou a métrica de tempo de ciclo no quadro kanban para mostrar quanto tempo os cartões estão em andamento, ou seja saíram da posição “doing” e chegaram até a posição “review”. Os cartões são atividades a serem desempenhadas pela equipe de projeto.

A equipe usa essa métrica principalmente para:

- Encontrar os cartões com tempo discrepante no gráfico durante a review do time, podendo ser uma ótima maneira de ajudar a equipe a descobrir (ou ser honesto) sobre os cartões que causaram este problema.
- Manter o tempo do ciclo estável ao longo do tempo, ou seja, para que a equipe tenha uma previsibilidade em um ciclo relativamente estável. Tentar manter os cartões abaixo da linha de 85% pode ajudar nisso.

A equipe em questão utiliza um gráfico com o tempo de ciclo que são construídas as tarefas, no eixo y do gráfico tem-se os dias em progresso e no eixo x as semanas, pois um tempo de ciclo é medido a cada uma semana completa de trabalho, gráfico ilustrado na figura 19.

Figura 19 - Tempo de ciclo tempo total das tarefas



Fonte: O autor (2020)

As linhas verdes, amarelas e vermelhas, correspondem a 50% das tarefas, 85% e 95%, respectivamente. O tempo do ciclo é geralmente relatado como uma média em vários cartões. Nesse caso, relatamos os percentis 50%, 85% e 95%. Se 50% percentil for 10 dias, isso significa que 50% dos cartões foram concluídos em 10 dias ou menos. Por observação do gráfico no time estudado temos 50% das tarefas concluídas em 4 dias.

Caso seja perguntado para essa equipe quanto tempo levará para concluir um cartão iniciado hoje, é possível dizer com 50% de certeza que será concluído em 4 dias ou menos. Se quem está perguntando precisa de algo melhor do que a probabilidade de 50% de chance de estar certo, é possível usar o percentil 85% ou 95% que para esta equipe é de 6 dias e 11 dias, respectivamente.

De acordo com a teoria das métricas da metodologia Kanban, conforme

Anderson (2011), idealmente o tempo de ciclo deve ser constante ou diminuir ao longo do tempo. Aumentar o tempo do ciclo significa que os cartões estão demorando mais, em média, para serem concluídos.

A falta de constância observada no gráfico pode ser explicada pelo fato da equipe de desenvolvimento ser composta por recursos humanos com pouca experiência em engenharia de software, portanto a previsibilidade de uma tarefa pode não ser tão certa devido à necessidade de aprendizado do conhecimento a cada novo desafio de desenvolvimento.

É possível perceber a partir de análises do tempo de ciclo, que em alguns pontos a equipe enfrentou tarefas com tempos altamente discrepantes, este tempo pode ser visto como discrepante pois foi usado como modelo uma equipe considerada de alta performance da empresa, em que 95% de chance da tarefa ficar pronta deve estar abaixo de 10 dias. As tarefas em questão que se encontram acima de 10 dias, devem ser pontos discutidos entre o time para endereçar ajuda, caso precise ou identificar aprendizados que possam ajudar em futuros projetos.

#### 4.4.2 Diagrama de fluxo acumulado

Outro indicador proposto é o diagrama de fluxo acumulado, o qual permite responder as seguintes perguntas:

1. O WIP (trabalho em andamento) está se acumulando ao longo do tempo?
2. O trabalho em alguma coluna individual está se acumulando ao longo do tempo?
3. O trabalho está sendo realizado a uma taxa constante?

A estrutura do gráfico representa no eixo horizontal um período de tempo (semanas, waves, etc), no caso estudado são as semanas e o vertical indica, de forma acumulada, o número de itens no processo para fazer, fazendo, em revisão e feito. Cada área pintada no gráfico está relacionada a uma etapa do fluxo de trabalho, sendo eles:

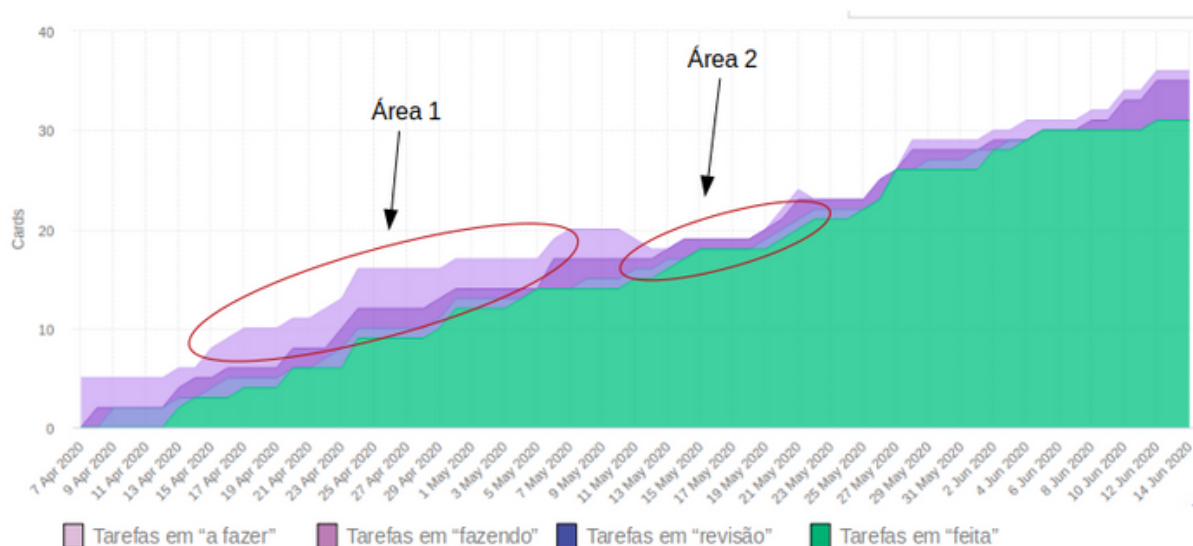
- Roxo claro: a fazer;
- Roxo escuro: fazendo;
- Azul: revisão; e,
- Verde feito.

As curvas são o número de itens acumulados em tais etapas.

No time em questão o diagrama foi utilizado para verificar se o time está trabalhando e realizando tarefas a uma taxa constante. Pois, idealmente se o gráfico parece com uma série de linhas aproximadamente paralelas, isso indica que o trabalho entra e sai de cada lista na mesma taxa. O gráfico do time pode ser

observado na figura 20.

Figura 20 - Diagrama de fluxo acumulado do projeto estudado



Fonte: O autor (2020)

Conforme pode ser visto na figura 20, o método Kanban segue o conceito de entrega contínua e portanto desde que não haja desequilíbrios no fluxo, são apresentadas, no diagrama, linhas que crescem quase que de forma paralela ao gráfico.

Não é incomum ver um padrão em que uma cor no gráfico apresenta uma faixa mais espessa, representado no gráfico como a área 1. Isto significa que os cartões estão acumulados nesta coluna do quadro kanban. Muitas vezes essa observação pode indicar que algumas etapas do processo não estão sendo executadas o mais rápido possível, pois, a taxa de tarefas deve ser constante.

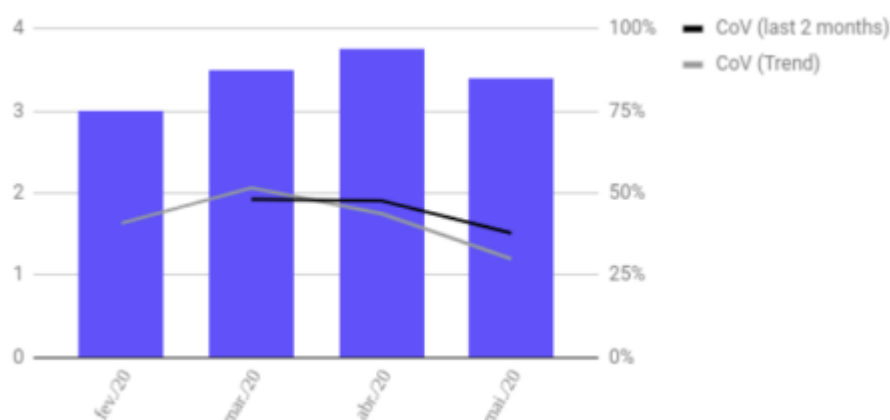
O oposto também pode ser observado na área 2 no diagrama exposto. Não existe uma faixa de área que representa a coluna " a fazer ", indicando que houve um período de escassez de tarefas a serem feitas, podendo trazer como consequência o ócio dos desenvolvedores da equipe, perdendo tempo preciosos de trabalho. Mais uma vez, cabe dentro das reuniões de review, o time solucionar e discutir as saídas de informações dos dados do gráfico.

#### 4.4.3 Throughput e coeficiente de variação

A métrica Throughput representa, a quantidade de cards do quadro kanban entregues num determinado período de tempo. Pode ser utilizada para responder a pergunta de previsibilidade da quantidade de cards que serão resolvidos a cada

semana. A figura 21 representa a métrica do time estudado em questão.

Figura 21 - Throughput e Coeficiente de variação



Fonte: O autor (2020)

Analisando a figura pode ser identificado que no mês de fevereiro a equipe entregou uma média de 3 tarefas por semana, no mês de março aproximadamente 3,5, no mês de abril 3,8 e por fim no mês de maio 3,4.

O seu coeficiente de variação de entregas semanais também é analisado, de acordo com o gráfico em fevereiro foi de 41%, em março 52%, em abril 44% e em maio 30%. Os valores indicam que, quanto mais baixo for o coeficiente de variação, maior será a previsibilidade se tem em relação ao próximo mês, quando for necessário fazer uma previsão de quantas tarefas podem ser entregues. Observando as métricas do time, pode-se concluir que ele ao longo dos meses se tornou mais previsível em relação a cadência dos trabalhos entregues.

#### 4.5 IMPLEMENTAÇÃO DE PEQUENAS MELHORIAS

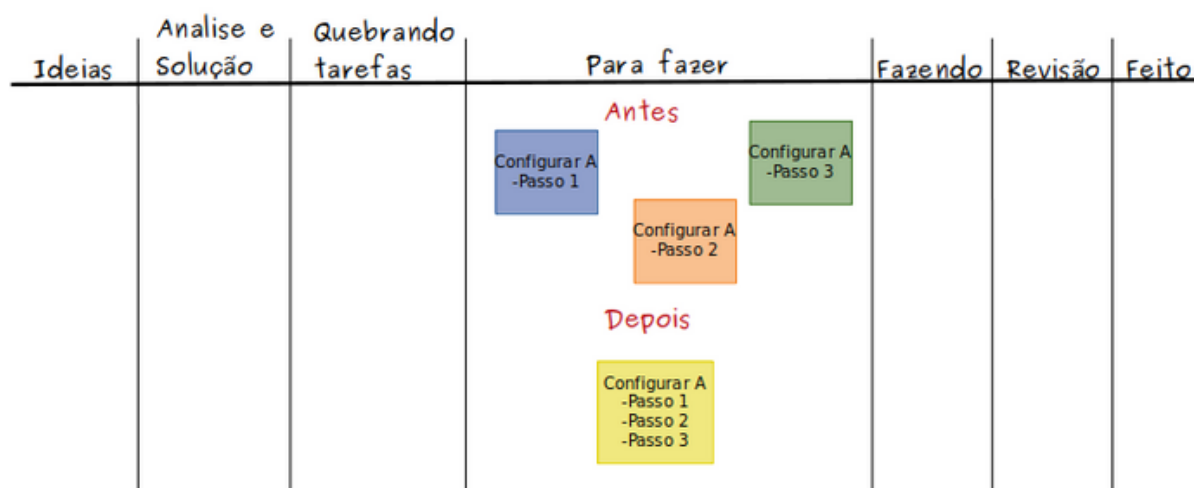
No decorrer da implantação do sistema Kanban, um fator de grande importância foi à confiança nos dados coletados ao longo da execução do trabalho. Um dos princípios da metodologia é melhorar colaborativamente e evoluir experimentalmente respaldado pelas métricas coletadas.

Portanto, o exemplo da métrica do diagrama de fluxo acumulado do time apresentado (figura 20), pode ser observado na área 1, que a lista de tarefas a serem feitas estava grande. Uma das soluções levantadas foi aumentar a complexidade das tarefas propostas para que pudesse diminuir o número de cartões parados na coluna.

Por exemplo, se um cartão representava o passo 1 de uma configuração A, outro cartão o passo 2 e assim por diante. Após a observação vista no diagrama,

experimentalmente colocamos toda a configuração em um só cartão. Conforme pode ser visto na figura 22.

Figura 22 - Implementação de melhoria na quebra de tarefas por observação



Fonte: O autor (2020)

O impacto da decisão de mudança poderá ser observado pelo acompanhamento das métricas propostas. Porém, a lição mais importante da atitude em mudar é um dos principais conceitos do Kanban, que traz agilidade em responder às mudanças que surgem ao longo da execução de tarefas, projetos e organização, formando um time que trabalha de forma mais eficiente.

#### 4.6 ANÁLISE DO QUESTIONÁRIO DE AVALIAÇÃO DA SOLUÇÃO PROPOSTA

Foi realizada uma pesquisa com 7 pessoas da equipe de desenvolvimento de software da empresa, incluindo (5) engenheiros de software, (1) *tech leader* e (1) gerente de produto, com o objetivo de avaliar a aplicação do método Kanban na empresa. O questionário utilizado na pesquisa está descrito no Apêndice 1, e os resultados podem ser observados no quadro 7. Para avaliação foi utilizada uma escala Likert, sendo 1 (Discordo totalmente), 2 (Discordo), 3 (Neutro), 4 (Concordo) e 5 (Concordo totalmente). Na tabela 8 é apresentada a média das respostas.

Quadro 8 - Afirmações e suas respectivas médias (continua)

Nº	Afirmação	Média
1	A abordagem Kanban permitem que as metas do projeto tenham enfoque nos resultados do negócio, potencializando atingir múltiplos critérios de sucesso.	4,29

Quadro 8 - Afirmações e suas respectivas médias (conclusão)

Nº	Afirmiação	Média
2	A abordagem Kanban auxiliar para que o plano do projeto tenha uma organização e um processo para atingir as metas esperadas e os resultados do negócio.	4,71
3	A abordagem Kanban permite que o planejamento seja realizado no início e reavaliado sempre que necessário.	4,43
4	Com o Kanban a abordagem gerencial é flexível, variável e adaptativa.	4,86
5	A abordagem Kanban facilita a atuação em projetos em que o trabalho é imprevisível, não mensurável, não linear e complexo	3,71
6	Com o Kanban é possível controlar, identificar mudanças no ambiente e ajustar o plano de projeto adequadamente.	4,43
7	Aplicação da abordagem Kanban é adaptável ao processo dependendo do tipo de projeto.	4,71
8	A abordagem Kanban possibilita a regulação do fluxo de itens globais com controle visual a fim de executar essas funções com precisão.	4,86
9	A abordagem Kanban possibilita a simplificação do trabalho administrativo dando autonomia das pessoas envolvidas no desenvolvimento do software.	4,57
10	Na sua opinião, a abordagem do Kanban na empresa permite: A transmissão de informação de forma organizada e rápida.	4,43
11	Na sua opinião, a abordagem do Kanban na empresa permite: A visualização do fluxo de trabalho.	4,71
12	Na sua opinião, a abordagem do Kanban na empresa permite: A limitação do trabalho em andamento.	4,29
13	Na sua opinião, a abordagem do Kanban na empresa permite: A definição de políticas de gestão de qualidade.	4,29
14	Na sua opinião, a abordagem do Kanban na empresa permite: O ajuste da cadência de trabalho.	4,86
15	Na sua opinião, a abordagem do Kanban na empresa permite: A mensuração do fluxo de trabalho.	4,86
16	Na sua opinião, a abordagem do Kanban na empresa permite: A definição de prioridades.	4,86
17	Na sua opinião, a abordagem do Kanban na empresa permite: A identificação de classes de serviço.	3,57
18	Na sua opinião, a abordagem do Kanban na empresa permite: A gerência de fluxo de informações e atividades.	4,29
19	Na sua opinião, a abordagem do Kanban na empresa permite: O estabelecimento de nível de serviço.	4,29
20	Na sua opinião, a abordagem do Kanban na empresa permite: O foco em melhoria contínua.	4,71

Fonte: O autor (2020)

Observando o quadro 8 pode-se concluir que apenas 10% das afirmativas

não ficaram entre concordo e concordo totalmente, são as afirmativas 5 e 17. É possível inferir de acordo com a afirmativa 17 que a metodologia kanban tem melhores resultados se o projeto for melhor discutido e delimitado na etapa mapeamento do problema. Pois, segundo o questionário, a metodologia não facilita a atuação em projetos em que o trabalho é imprevisível, não mensurável, não linear e complexo. Além disso, ainda segundo os resultados do questionário, a forma como o método é utilizado na empresa não expõe com tanta clareza as classes dos serviços realizados no fluxo de trabalho. Uma possível proposta de melhoria para este problema seria criar uma legenda que indicasse a classe do serviço, a ser feita em cada tarefa.

Entretanto, de forma geral é possível afirmar que a metodologia kanban na prática aplicada na empresa, iguala as expectativas do método teórico descrito no Capítulo 2.



## 5 CONCLUSÃO

Neste trabalho foi apresentado um modelo de gerenciamento de projetos baseado na abordagem Kanban. A abordagem possibilita a coleta de métricas ágeis por meio da integração com um quadro visual, virtual e online kanban, com enfoque nos indicadores necessários dentro do processo de software de equipes em ambientes de desenvolvimento ágil.

Primeiramente realizou-se um estudo da literatura na área de desenvolvimento de software. Esse estudo envolveu os conceitos básicos e genéricos das etapas de um projeto em engenharia de software.

Nesse contexto de desenvolvimento de software, foi realizado um estudo sobre a metodologia tradicional de gerenciamento de projetos (visão do PMBoK), seguido das metodologias ágeis de desenvolvimento de software. O enfoque principal deste capítulo foi o documento do manifesto ágil.

Na sequência foi detalhado o método ágil Scrum, que se destaca dentre todas existentes por ter uma abordagem enxuta de gerenciamento de projetos. De qualquer forma, o principal enfoque foi a metodologia Kanban que visa otimizar o fluxo do trabalho e tem o objetivo de minimizar a resistência à mudança e facilitá-la, com foco no trabalho que atinge as necessidades do cliente.

Na sequência foi realizado um estudo de caso na empresa, o qual corroborou para o desenvolvimento da proposta do modelo.

O modelo proposto é composto das etapas de definição do problema a ser tratado definição de cerimônias, coleta e análise de métricas. De forma iterativa, as métricas coletadas permitem implementar mudanças ao longo do desenvolvimento do software, contribuindo para a agilidade do processo

Desta forma, entende-se que objetivo geral do trabalho de propor uma metodologia, composta por um conjunto de métodos e ferramentas do gerenciamento ágil de projetos em uma empresa de software, permitindo o registro do conhecimento com práticas de controle executada pela gerência da engenharia de software foi atingido.

Além disso, os objetivos específicos de: i) Revisar bibliograficamente as abordagens clássicas e ágeis; ii) Identificar o modelo, prática, técnicas e ferramentas de gestão do gerenciamento ágil; iii) Compreender o processo de desenvolvimento de software, e suas diferentes etapas; iv) Avaliar o modelo teórico através de um estudo de caso foram realizados e descritos com sucesso.

Este trabalho traz conteúdo relevante para a academia na área de desenvolvimento de software, gerenciamento de projetos na forma tradicional e ágil e aplicação de uma metodologia de gerenciamento que apoie o gestor no processo de desenvolvimento de software, além de possuir relevante pesquisa bibliográfica.

Quanto a limitação do estudo, sugere-se as seguintes possibilidades para trabalhos futuros:

- Aplicação de outras ferramentas gerenciamento de projetos, a fim de possibilitar uma comparação entre elas;
- Estudo da implementação do software;
- Aplicar o questionário para avaliar a satisfação dos clientes finais com o resultado final do software.

Por fim, ressalta-se o valor do trabalho de conclusão de curso para a formação do engenheiro. A partir deste trabalho foi possível aprimorar o conhecimento teórico nas áreas de interesse, assim como um maior entendimento da realidade das empresas de tecnologia.

## REFERÊNCIAS

- AGILE. **Agile manifesto**. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 27 Jul. 2020.
- AMANCIO, Fernando Stapf *et al.* **Como calcular o custo de uma alteração de escopo?**. **pmkb**. 2015. Disponível em: <https://pmkb.com.br/artigos/como-calculiar-o-custo-de-uma-alteracao-de-escopo/>. Acesso em: 8 Fev. 2020.
- AMARAL, Daniel Capaldo *et al.* **Gerenciamento Ágil de Projetos**: aplicação em produtos inovadores. 1. Ed. São Paulo: Saraiva, 2011.
- AMARAL, Daniel. **Metodologias Ágeis em Gerenciamento de Projetos**. 2012. Disponível em: <https://pt.slideshare.net/amaraldn/magp-2011-versaoweb>. Acesso em: 27 Jul. 2020.
- ANDERSON, DAVID. **Kanban**: Mudança Evolucionária de Sucesso Para Seu Negócio de Tecnologia. Blue Hole Press, 2011.
- BALDO, Rodrigo *et al.* APLICAÇÃO DA METODOLOGIA SCRUM EM UM ESTUDO DE CASO DE ENGENHARIA. **Produção online**, v. 19, n. 3, p. 856-875, 2019.
- BECK, Kent *et al.* **Manifesto para Desenvolvimento Ágil de Software**. 2001. Disponível em: <http://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 25 Jan. 2020.
- BERNARDO, Kleber. **Kanban: Do início ao fim!**. **Cultura ágil**. 2014. Disponível em: <https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/>. Acesso em: 27 Jul. 2020.
- COSTA, Filipe Sousa . **Fusão de um processo tradicional de desenvolvimento de software com uma metodologia ágil: um estudo de caso**.. UBERLÂNDIA, 2017. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) - Universidade Federal de Uberlândia, UBERLÂNDIA.
- CRUZ, Fabio. **Scrum, PMBOK® E Kanban, Um Modelo Híbrido De Perfeita Sinergia**. 2015. Disponível em: <https://pt.slideshare.net/ProjetosETi/scrum-pmbok-e-kanban-um-modelo-hbrido-de-perfeita-sinergia>. Acesso em: 27 Jul. 2020.
- DEMING, William Edwards. **O método Deming de Administração**. 5. ed. São Paulo: Marques Saraiva, 1989.
- DIGITAL.AI. **Agile Techniques employed. State of agile**. 2018. Disponível em: <https://www.stateofagile.com/#ufh-i-423641583-12th-annual-state-of-agile-report/473508>. Acesso em: 27 Jul. 2020.
- DINGSØYR, T *et al.* A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, v. 85, n. 6, p. 1213–1221, 2012.
- FARINAZZO, Raphael. **Por que nossos Product Owners viraram Product Managers (e qual a diferença)**.

<http://shipit.resultadosdigitais.com.br/blog/product-owners-viraram-managers/>. 2016. Disponível em: . Acesso em: 27 Jul. 2020.

FERENHOF, Helio A.; FORCELLINI, Fernando A.. Lições Aprendidas em um Projeto Mal Sucedido. *In: VI CONGRESSO BRASILEIRO DE GERENCIAMENTO DE PROJETOS DO PMI*. 2011, Fortaleza: Anais VI Congresso Brasileiro de Gerenciamento de Projetos do PMI, 2011.

FUNDAÇÃO DE AMPARO À PESQUISA DO ESTADO DE SÃO PAULO. **DADOS:** Formação de engenheiros e profissionais afins. **FAPESP**. 2018. Disponível em: [https://revistapesquisa.fapesp.br/wp-content/uploads/2018/05/011\\_Dados\\_267.pdf](https://revistapesquisa.fapesp.br/wp-content/uploads/2018/05/011_Dados_267.pdf). Acesso em: 27 Jul. 2020.

GENARI, João; FERRARI, Fabiano. Times de alto desempenho no contexto das metodologias Scrum e Kanban. **T.I.S.**, 2015.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002.

GOELZER, Viviane *et al.* Análise de um Processo de Inovação a partir da Ótica de Gestão de Projetos. **Revista de Gestão e Projetos**, v. 5, n. 2, p. 78-89, 2014.

HIGHSMITH, Jim; COCKBURN, Alistair. Agile software development: the business of innovation. **Computer**, v. 34, p. 120 - 127, 10 2001.

IKEDA, P. **A falta que bons engenheiros fazem**. **Revista Exame**. 2012. Disponível em: <https://exame.abril.com.br/revista-exame/a-falta-que-eles-fazem/>. Acesso em: 26 Jan. 2020.

INSTITUTO DE ESTUDOS PARA O DESENVOLVIMENTO INDUSTRIAL. **Carta IEDI:** Edição 858. **IEDI**. 2018. Disponível em: [https://www.iedi.org.br/cartas/carta\\_iedi\\_n\\_858.html](https://www.iedi.org.br/cartas/carta_iedi_n_858.html). Acesso em: 26 Jan. 2020.

KERZNER, Harnold. **Gestão de Projetos:** as Melhores Práticas. Porto Alegre: Bookman, v. 2, 2007.

LEITE, Madalena *et al.* Aplicação do sistema kanban no transporte de materiais na construção civil. *In: ENEGEP*. 2004, Florianópolis, 2004.

MACHADO, Rafaela Sano. **DESENVOLVIMENTO ÁGIL DE SOFTWARE: UMA REVISÃO SISTEMÁTICA DA LITERATURA**. Brasília, 2018. Trabalho de Conclusão de Curso (Engenharia de Produção) - Universidade de Brasília, Brasília, 2018.

MCCONNELL, S. Rapid development: taming wild software schedules. *In: MICROSOFT PRESS*. 1996, Redmond, WA, Estados Unidos, 1996.

NETO, Paulo. **IMPLEMENTAÇÃO DE UM MÓDULO PARA INTEGRAÇÃO DA ABORDAGEM USE-CASE 2.0 COM UM SISTEMA DE GERENCIAMENTO DE PROJETOS**. Florianópolis, 2017. Trabalho de Conclusão de Curso (Ciências da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2017.

PEREIRA, Vinicius. **Funil de vendas:** o que é, para que serve e como montar um.

**Blog de Marketing Digital de Resultados.** 2019. Disponível em: <https://resultadosdigitais.com.br/blog/o-que-funil-de-vendas/>. Acesso em: 27 Jul. 2020.

PINTO, Fabrícia Avendano; FRANK, Alejandro Gérman; PAULA, Istefani Carisio de. *In: 8º CONGRESSO BRASILEIRO DE GESTÃO DE DESENVOLVIMENTO DE PRODUTO.* 2011, Porto Alegre, 2011.

PINTO, HUDSON LAMOUNIER. **Atividades básicas ao processo de desenvolvimento de Software.** DevMedia. 2007. Disponível em: <https://www.devmedia.com.br/atividades-basicas-ao-processo-de-desenvolvimento-de-software/5413>. Acesso em: 27 Jul. 2020.

PMI. **A guide to the project management body of knowledge (PMBOK Guide).** 5. ed. Project Management Institute, 2013.

PRESSMAN, Roger S.. **Engenharia de Software: Uma Abordagem Profissional.** 7. ed. New York: The McGraw-Hill Companies, Inc, 2011.

RADIGAN, Dan. **Kanban: Como a metodologia Kanban é aplicada ao desenvolvimento de software.** Atlassian. 2019. Disponível em: <https://www.atlassian.com/br/agile/kanban>. Acesso em: 29 Abr. 2020.

REIS, P. Mercado visto de dentro. **Revista Construção Mercado**, v. 138, 2013.

SANTOS, José Luiz dos; SCHMIDT, Paulo. **Avaliação de Empresas: Foco na Análise de Desempenho para o Usuário Interno.** 1. ed. Editora Atlas S.A, 2006.

SCHINDLER, Martin; EPPLER, Martin J.. Harvesting project knowledge: a review of project learning methods and success factors. **Journal of Project Management**, v. 21, p. 219-228, 2003.

SCHUCH, CLAUDIO GARGIONI. **PROPOSTA DE UM MODELO PARA O PLANEJAMENTO ÁGIL DO PROJETO DE PRODUTOS.** Florianópolis, 2009. Dissertação (MESTRE EM ENGENHARIA MECÂNICA) - Universidade Federal de Santa Catarina, Florianópolis, 2009.

SCHWABER, Ken; SUTHERLAND, Jeff. **Um Guia definitivo para o Scrum: As regras do Jogo.** 2017.

SCRUM.ORG. **The Kanban Guide for Scrum Teams.** 2018. Disponível em: [https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-04/2018%20Kanban%20Guide%20for%20Scrum%20Teams\\_0.pdf](https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-04/2018%20Kanban%20Guide%20for%20Scrum%20Teams_0.pdf). Acesso em: 26 Abr. 2020.

SHENHAR, A; DVIR, D. **Toward a typological theory of project management.** 4. ed. Research Policy, v. 25, 1996.

SILVA FILHO, A. M.. Software Everywhere: sobre a demanda de software e da Engenharia de Software. **Revista Espaço Acadêmico**, v. 15, n. 172, p. 01-04, 2015.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e**

**Elaboração de Dissertação** . 4. ed. Florianópolis , 2005.

SLACK, Nigel *et al.* **Administração da produção**. 3ª. ed. São Paulo: Atlas, 2009.

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

TAKEUCHI, H; NONAKA, I. **The new new product development game**. **Harvard Business Review**. 1986. Disponível em: <https://hbr.org/1986/01/the-new-new-product-development-game>. Acesso em: 27 Jul. 2020.

WILLIAMS, Laurie; COCKBURN, Alistair. **Agile Software Development: It's About Feedback and Change**. 2003. Disponível em: <https://collaboration.csc.ncsu.edu/laurie/Papers/feedbackAndChange.pdf>. Acesso em: 27 Jul. 2020.



4. Com o Kanban a abordagem gerencial é flexível, variável e adaptativa.

*Marcar apenas uma oval.*

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

5. A abordagem Kanban facilita a atuação em projetos em que que o trabalho é imprevisível, não mensurável, não linear e complexo \*

*Marcar apenas uma oval.*

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

6. Com o Kanban é possível controlar, identificar mudanças no ambiente e ajustar o plano de projeto adequadamente. \*

*Marcar apenas uma oval.*

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

7. Aplicação da abordagem Kanban é adaptável ao processo dependendo do tipo de projeto.

*Marcar apenas uma oval.*

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente





10. Na sua opinião, a abordagem do Kanban  permite: \*

*Marcar apenas uma oval por linha.*

	Discordo totalmente	Discordo	Indiferente	Concordo	Concordo totalmente
A transmissão de informação de forma organizada e rápida.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A visualização do fluxo de trabalho.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A limitação do trabalho em andamento.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A definição de políticas de gestão de qualidade.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O ajuste da cadência de trabalho.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A mensuração do fluxo de trabalho.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A definição de prioridades.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A identificação de classes de serviço.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A gerência de fluxo de informações e atividades.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O estabelecimento de nível de serviço.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O foco em melhoria contínua.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>