



UNIVERSIDADE DO SUL DE SANTA CATARINA
JOÃO VITOR SANTOS DA SILVA

**CUSTOMIZAÇÃO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE À
LUZ DOS PRECEITOS DAS METODOLOGIAS ÁGEIS**

Palhoça
2021

JOÃO VITOR SANTOS DA SILVA

**CUSTOMIZAÇÃO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE À
LUZ DOS PRECEITOS DAS METODOLOGIAS ÁGEIS**

Este trabalho de conclusão de curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Graduação em Sistemas de informação da Universidade do Sul de Santa Catarina.

Orientadora: Vera Rejane Niedersberg Schuhmacher, Dra.

Palhoça

2021

JOÃO VITOR SILVA DOS SANTOS

**CUSTOMIZAÇÃO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE À
LUZ DOS PRECEITOS DAS METODOLOGIAS ÁGEIS**

Este trabalho de conclusão de curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Graduação em Sistemas de informação da Universidade do Sul de Santa Catarina.

Palhoça, 21 de junho de 2022.

Profa. Orientadora Vera Rejane Niedersberg Schuhmacher, Dra.
Universidade do Sul de Santa Catarina

Profa. Maria Inés Castiñeira, Dra.
Universidade do Sul de Santa Catarina

Prof. Aran Bey Tcholakian Morales, Dr.
Universidade do Sul de Santa Catarina

Agradeço aos meus pais e avós pelo apoio de sempre e dedico este trabalho a eles. Esta monografia é a prova de que todo o investimento na minha educação e o carinho e cuidado que tiveram comigo valeram a pena.

AGRADECIMENTOS

Não poderia iniciar estes agradecimentos sem falar dos meus pais e avós, a quem sou eternamente grato, gratidão essa não apenas pela força nos momentos difíceis, mas também pelo incentivo durante toda a minha trajetória acadêmica. Vocês foram o meu pilar motivador durante todo o desenvolvimento desta pesquisa.

Agradeço também aos amigos que estiveram do meu lado no decorrer do curso e que vivenciaram muitas situações e momentos difíceis junto comigo. Vocês tornaram tudo mais leve, pois eu sabia com quem sempre contar.

Aos meus líderes agradeço todo o apoio concedido durante o trabalho e, principalmente, sou grato por me inspirarem na escolha do tema para esta monografia.

À minha professora orientadora, agradeço por aceitar me conduzir nessa jornada, por todo o apoio e pelas horas dedicadas a esta pesquisa. Professora Vera, sem você nada disso seria possível.

Por fim, a todos os que de alguma maneira contribuíram com a minha formação. A vocês, o meu muito-obrigado.

“Eu posso não ter ido para onde eu pretendia ir, mas eu acho que acabei terminando onde eu pretendia estar.” (Douglas Adams, 1988)

RESUMO

Esta monografia compreende a pesquisa e a análise de adequação da metodologia de desenvolvimento de software considerando os preceitos das metodologias ágeis para uma empresa de médio porte, doravante chamada de empresa Beta. Essa empresa possui um time de tecnologia enxuto com um mercado promissor, contudo necessita aumentar a eficiência e a qualidade dos processos envolvidos na sua produção de software. A pesquisa proposta neste trabalho se enquadra como aplicada, pois tem como finalidade o desenvolvimento voltado para uma aplicação prática, buscando solucionar um problema específico. O principal objetivo do projeto é propor uma metodologia customizada ao cenário atual da empresa Beta, à luz dos preceitos, das estratégias e das técnicas sugeridas pelas metodologias ágeis. As recomendações propostas envolvem e/ou se originam dos modelos Scrum, XP e Kanban. As recomendações de customização e o redesenho do processo foram verificados por três componentes do time de desenvolvimento da empresa Beta. A análise dos resultados possibilita compreender a aderência da proposta, tendo em vista que foi de concordância entre todos os entrevistados que o processo proposto está aderente à visão de futuro da empresa, ao planejamento estratégico desta e, principalmente, ao que gera valor à companhia. Dessa forma, entende-se que o processo proposto trará ganhos ao time de tecnologia da empresa Beta.

Palavras-chave: Metodologia. Ágil. Scrum. Kanban, XP.

ABSTRACT

This thesis involves the research and analysis of the adequacy of the software development methodology considering the precepts of agile methodologies for a medium-sized company, henceforth called Beta company. This company has a lean technology team with a promising market; however, it needs to increase the efficiency and quality of the processes involved in the software production. The research proposed in this work is classified as applied, as it aims to develop a practical application, seeking to solve a specific problem. The main objective of the project is to propose a customized methodology to the current scenario of the company Beta, in the light of the precepts, strategies and techniques suggested by agile methodologies. The proposed recommendations involve and/or originate from the Scrum, XP, and Kanban models. The customization recommendations and the process of redesign were verified by three components of the Beta company's development team. The analysis of the results makes it possible to understand the adherence of the proposal, considering that all the interviewees agreed that the proposed process matches to the company's vision of the future, to its strategic planning and mainly to what adds value for the company. Thus, it is understood that the proposed process will bring gains to the technology team of company Beta.

Keywords: Methodology. Agile. Scrum. Kanban, XP.

LISTA DE ILUSTRAÇÕES

Figura 1 - Backlog	19
Figura 2 - Quadro Kanban.....	26
Figura 3 - Processo de desenvolvimento de projeto	31
Figura 4 - Tela de processo para desenvolvimento de projeto.....	32
Figura 5 - Tela de Kanban para desenvolvimento de projeto	37

SUMÁRIO

1	INTRODUÇÃO	11
1.1	PROBLEMÁTICA.....	12
1.2	OBJETIVOS	12
1.2.1	Objetivo geral	13
1.2.2	Objetivos específicos	13
1.3	JUSTIFICATIVAS	13
1.4	ESTRUTURA DA MONOGRAFIA	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	MODELOS ÁGEIS.....	15
2.1.1	Manifesto Ágil	16
2.2	SCRUM	16
2.2.1	Característica do Scrum	17
2.2.2	Papéis no SCRUM	17
2.2.3	Artefatos do Scrum	18
2.2.4	Cerimônias do Scrum	20
2.2.5	Vantagens do Scrum	21
2.3	EXTREME PROGRAMMING (XP)	21
2.3.1	Vantagens da XP	24
2.4	KANBAN.....	25
2.4.1	Vantagens do Kanban	26
2.5	CONSIDERAÇÕES SOBRE O CAPÍTULO	27
3	ASPECTOS METODOLÓGICOS	28
3.1	ETAPAS METODOLÓGICAS DE PESQUISA.....	28
3.1.1	Fundamentação teórica	29
3.1.2	Modelo do processo atual da empresa	29
3.1.3	Identificação das fragilidades e possibilidades de melhoria do processo atual	29
3.1.4	Customização do processo a partir dos modelos ágeis	29
3.1.5	Verificação do processo customizado	30
4	DESENVOLVIMENTO	31
4.1	MODELO DO PROCESSO ATUAL DA EMPRESA BETA (AS-IS)	31
4.2	SUGESTÕES DE MELHORIA DO PROCESSO ATUAL.....	33

4.2.1	Conceitos do Scrum e visão pontual dos processos da empresa Beta	33
4.2.1.1	Reunião de planejamento da Sprint.....	34
4.2.1.2	Daily Sprint	34
4.2.1.3	Revisão da Sprint	34
4.2.2	Conceitos da XP	35
4.2.2.1	Testes de usuários	35
4.2.2.2	Código coletivo	35
4.2.2.3	Refactoring	35
4.2.2.4	Programação em pares	36
4.2.3	Conceitos do Kanban.....	36
5	VERIFICAÇÃO DO PROCESSO (TO-BE).....	38
5.1	RESULTADOS DA VERIFICAÇÃO DO PROCESSO (TO-BE)	38
5.1.1	Primeiro entrevistado	38
5.1.2	Segundo entrevistado	39
5.1.3	Terceiro entrevistado	40
5.2	CONCLUSÕES DAS ENTREVISTAS	41
6	CONCLUSÃO E TRABALHOS FUTUROS	42
	REFERÊNCIAS BIBLIOGRÁFICAS	43

1 INTRODUÇÃO

O termo “qualidade de software” surgiu aproximadamente na década de 1980, quando a necessidade de fazer entregas cada vez mais assertivas e com menos retrabalho se destacou em um mercado relativamente novo mas promissor, e que já se apresentava competitivo – o desenvolvimento de software. A definição de qualidade de software é um termo que pode possuir diferentes interpretações. Segundo Pressman (1995, p. 441), qualidade de software é:

[...] a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais.

Alguns dos pontos que definem se um software tem qualidade são: a - possuir controle sobre os prazos e o custo do software; b - atender aos requisitos definidos no escopo do projeto; c - reduzir erros nas entregas, d - reduzir retrabalho; e - atender a mudanças que possam ocorrer durante o desenvolvimento.

As metodologias ágeis surgem no universo de desenvolvimento de sistemas computacionais com o intuito de apoiar as questões que envolvem aspectos intrinsecamente ligados à qualidade do processo e do produto de desenvolvimento de software. São padrões de organização para o desenvolvimento de projetos (normalmente voltados a software) que têm como objetivo a entrega de um produto ou serviço, de forma que as entregas deste sejam feitas parcialmente por interações. Em cada uma dessas interações o cliente acompanha o desenvolvimento do software, evitando que sejam desenvolvidas funcionalidades a mais ou a menos no projeto. As ações de planejamento com a coparticipação do cliente reiteram a premissa de eficiência, rapidez, qualidade e custos mais assertivos, evitando retrabalho.

As metodologias ágeis surgiram em 2001, ano em que um grupo de 17 pesquisadores – entre eles os criadores do Scrum e do Feature-Driven Development (FDD) – reuniu-se na estação de esqui Snowbird, nas montanhas nevadas de Utah (EUA), para elaboração de um manifesto que prezava por princípios essenciais no desenvolvimento de software, tais como experiência do cliente, agilidade e, principalmente, qualidade nas entregas (NOLETO, 2020). Esse documento foi chamado de Manifesto Ágil e até os dias atuais é a base de fundamentos para todas

as metodologias ágeis criadas a partir de então. O manifesto conta com 4 valores e 12 princípios que devem ser incluídos nas rotinas das empresas que desejam se tornar ágeis.

A qualidade do processo de desenvolvimento de software bem como a qualidade da gerência do projeto são fundamentais para o sucesso da empresa e para a manutenção da competitividade no mercado de Tecnologia da Informação (TI). Decisões assertivas nesse sentido são complexas, pois as empresas desenvolvedoras diferem entre si quando o assunto é a forma com que ocorre o seu processo de desenvolvimento de software bem como as características das suas equipes e da sua gestão.

1.1 PROBLEMÁTICA

Este projeto está fundado no estudo de caso da empresa Beta, doravante assim chamada para que se mantenha o seu anonimato. A empresa Beta atua no ramo varejista do mercado tático e militar, e teve um crescimento exponencial nos últimos anos, o que triplicou o seu quadro de funcionários, assim como o seu lucro bruto. Com isso, a empresa necessitou aumentar o seu time de tecnologia, de forma a suprir demandas internas, desenvolver softwares, trabalhar seus dados empresariais e, principalmente, tornar-se um setor estratégico para que a sua liderança no segmento tático possa ser mantida.

Atualmente o setor de TI da empresa não conta com uma metodologia de trabalho formalizada. A entrega dos projetos é realizada com base em uma rotina tradicional ao final do desenvolvimento apoiado no modelo cascata, uma metodologia antiga de gerenciamento de projetos. O cliente, ansioso pelo acompanhamento do projeto, acaba por vivenciar uma grande insatisfação no momento da entrega, visto perceber que em muitas situações houve mudança de requisitos. Isso acarreta um índice de retrabalho extremamente alto, sendo necessário muitas vezes o reinício do projeto a partir de novos requisitos que foram levantados durante o desenvolvimento.

1.2 OBJETIVOS

Nas seções que se seguem, são apresentados os objetivos geral e específicos que norteiam a presente monografia.

1.2.1 Objetivo geral

O principal objetivo do projeto é propor uma metodologia customizada ao cenário atual da empresa Beta à luz dos preceitos, das estratégias e das técnicas propostas pelas metodologias ágeis.

1.2.2 Objetivos específicos

Além do objetivo principal, o projeto propõe o alcance dos seguintes objetivos específicos:

- obter maior qualidade no produto a ser entregue;
- controlar e registrar as entregas feitas pelo time de TI;
- gerar visibilidade para os demais setores das entregas feitas pelo time de TI;
- reduzir o retrabalho do time de TI;
- reduzir custos desnecessários gerados pelo retrabalho;
- pesquisar e identificar diferentes metodologias ágeis e estratégias a fim de melhorar e customizar o processo atual da empresa;
- desenvolver um processo de atendimento do setor de TI consistente para todas as áreas; e
- promover a qualidade do processo e o rastreamento de situações de risco.

1.3 JUSTIFICATIVAS

Os benefícios provenientes da utilização de modelos ágeis são visíveis em qualquer empresa que faz a implantação de modelos que promovam a perspectiva ágil. A otimização de processos e a diminuição da burocracia são metas buscadas incessantemente no mundo dos negócios. Dessa forma, empresas que seguem os princípios estipulados no Manifesto Ágil possuem benefícios de agilidade. Prazos de entrega reduzidos permitem ao cliente ter maior interação com o projeto, e, com isso, erros de desenvolvimento podem ser evitados. Com a decomposição do problema em etapas, a previsão e a mensuração de custo e de prazos mais assertivos tornam-se

mais simples, pois o orçamento é feito em microetapas. Também há redução drástica de riscos no desenvolvimento, uma vez que o cliente está em constante acompanhamento do projeto.

Sendo assim, a principal justificativa para esta monografia é a reestruturação do modelo de trabalho adotado atualmente pelo setor de TI da empresa Beta, visto que esse modelo apresenta contradições que prejudicam as entregas, os prazos e a lucratividade do processo em seu conjunto.

1.4 ESTRUTURA DA MONOGRAFIA

A presente monografia compreende cinco capítulos, cada um deles eles descrito resumidamente a seguir.

O capítulo 1 traz a temática, os objetivos, a justificativa e o problema a ser solucionado.

O capítulo 2 aporta o referencial teórico, contendo temáticas que sustentam o desenvolvimento da monografia, tais como modelos ágeis, Scrum, Extreme Programming, Kanban e Feature-Driven Development.

No capítulo 3 é explicitada a proposta metodológica do trabalho.

O capítulo 4 versa sobre o processo de adequação e sobre as escolhas das recomendações inferidas como apropriadas para a customização do processo e para a sua qualificação.

O capítulo 5 apresenta as considerações finais do autor e os trabalhos futuros para interessados no tema.

2 FUNDAMENTAÇÃO TEÓRICA

No capítulo de fundamentação teórica serão apresentadas leituras e reflexões acerca das seguintes temáticas: modelos ágeis, Scrum, Extreme Programming e Kanban.

2.1 MODELOS ÁGEIS

As metodologias ágeis surgiram em meados de 1990, em caminho adverso às pesadas metodologias de desenvolvimento utilizadas na época. Os modelos então existentes prezavam a documentação, caracterizada por processos microgerenciados extremamente burocráticos, o que por muitas vezes dificultava o desenvolvimento dos softwares. Em sua grande maioria, essas metodologias de desenvolvimento se baseavam no modelo cascata, que tinha como atributo entregar o projeto ao solicitante apenas quando todas as funcionalidades já estivessem prontas, o que gerava um grande índice de retrabalho e, conseqüentemente, impactava na satisfação do cliente.

Com um mercado cada vez mais competitivo como o da tecnologia, profissionais e pesquisadores iniciaram propostas de métodos com o intuito de promover um processo de desenvolvimento mais célere, dando início ao movimento das metodologias ágeis. A fim de discutir ideias e experiências sobre formas de melhorar o desempenho de seus projetos, um grupo de pesquisadores se reuniu na estação de esqui Snowbird, em Utah (EUA), durante um final de semana. Entre esses pesquisadores estavam Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas (WARD CUNNINGHAM, c2001).

Durante esse encontro, constatou-se que cada um dos participantes utilizava um conjunto diferente de boas práticas para as suas metodologias, porém todos concordaram que existiam princípios básicos para um projeto ter sucesso. A partir dessa constatação, os pesquisadores decidiram redigir um documento com esses princípios, o qual foi batizado de “Manifesto Ágil” e assinado por todos os que participaram do encontro. Até os dias atuais, o manifesto serve como base referencial para metodologias ágeis.

2.1.1 Manifesto Ágil

O Manifesto Ágil criado pelos pesquisadores apontava a emergência na construção de softwares de qualidade considerando o uso de hardwares mais avançados, linguagens de programação, ambientes de desenvolvimento e necessidades organizacionais cada vez mais arrojadas na automação de processos em todos os setores da sociedade. Composto por 12 princípios, o manifesto norteia qualquer metodologia ágil de desenvolvimento (NOLETO, 2020, n.p):

1. priorizar a satisfação de usuários e usuárias com uma entrega contínua de produtos de alto valor agregado e dentro (ou antes) dos prazos;
2. ser flexível em relação a mudanças, ainda que elas ocorram nas etapas finais do processo de desenvolvimento;
3. sempre buscar entregar softwares que sejam funcionais;
4. manter um time integrado e que colabora em busca de objetivos em comum ao longo de todo o projeto, ou seja, ter uma liderança que trabalha ao lado da equipe de desenvolvimento;
5. a liderança deve manter as equipes motivadas e com o suporte necessário para executar o seu trabalho da melhor forma possível e sempre com muita autonomia;
6. conversar frente a frente deve ser visto como a melhor forma de transmitir informações para quem estiver no projeto;
7. ter um software funcional é a principal forma de avaliar se há progresso;
8. entender que um projeto pautado pela metodologia ágil está em contínua evolução, portanto, a pessoa que estiver envolvida precisa ter bastante atenção;
9. saber que agilidade se ganha com uma atenção contínua à técnica e um bom design;
10. focar na simplicidade envolve maximizar a quantidade de esforço que não é feito;
11. ter uma equipe em que todas as pessoas podem se planejar de modo autônomo garante melhores resultados ao final do projeto;
12. reunir as equipes regularmente para alinhar expectativas, trocar ideias e avaliar qual o melhor caminho para seguir em frente.

2.2 SCRUM

Scrum é uma metodologia ágil que possui grande influência das boas práticas adotadas pela indústria japonesa, principalmente pelas companhias Honda e Toyota. Essas boas práticas defendem que projetos que usam equipes pequenas e multidisciplinares produzem melhores resultados quando comparadas a grandes equipes com papéis definidos para cada um dos indivíduos.

De acordo com Sbrocco e Macedo (2012, p. 159),

A denominação dessa metodologia surgiu da associação dessas equipes de projeto altamente eficazes com uma típica formação do evento esportivo Rugby denominada Scrum. No Rugby, essa formação é utilizada após determinado incidente ou quando a bola sai de campo, ou seja, é utilizada para reiniciar o jogo, reunindo todos os jogadores. O uso dessa terminologia pareceu adequado porque no Rugby cada time age em conjunto, como uma unidade integrada, cada membro desempenha um papel específico e todos se ajudam em busca de um benefício comum.

2.2.1 Característica do Scrum

Segundo Jeff Sutherland e Ken Schwaber, criadores da metodologia, o Scrum se fundamenta em 6 principais características: (1) flexibilidade dos resultados, (2) flexibilidade dos prazos, (3) times pequenos, (4) revisões frequentes, (5) colaboração e (6) orientação a objetos (SBROCCO; MACEDO, 2012). Projetos que utilizam Scrum como metodologia baseiam-se em três papéis, três artefatos e quatro cerimônias:

Três papéis: o Dono do Produto (DP), responsável pelo valor de negócio do produto, o Scrum Master, que serve aos outros papéis com processos e iniciativas para melhorar o trabalho, e a Equipe de Desenvolvimento, que se auto-organiza para entregar valor por meio do desenvolvimento de software com qualidade.

Três artefatos: o Backlog do Produto, que lista a visão atualizada dos requisitos desejados para o produto, o Incremento do Produto, representado pelas funcionalidades de software pronto, e o Backlog da Sprint, que representa o planejamento estratégico e tático da próxima Sprint em um nível mais micro.

Quatro cerimônias: a reunião de planejamento da Sprint, na qual será conhecida e planejada a meta de uma Sprint; as Scrum Diárias (Reuniões Diárias), que elevam o nível de auto-organização da Equipe de Desenvolvimento; a Revisão da Sprint, em que produto e projeto serão inspecionados e adaptados; e a Retrospectiva da Sprint, momento para inspeção e adaptação dos processos empíricos (PRIKLADNICKI; WILLI; MILANI, 2014, p. 46).

2.2.2 Papéis no SCRUM

O Scrum, diferentemente de outras metodologias ágeis, não possui apenas um responsável pelo projeto, mas sim três. O dono do produto (DP) é responsável por assegurar que o trabalho executado pela equipe de desenvolvimento gere o retorno estimado no escopo do projeto. Segundo Prikladnicki, Willi e Milani (2014, p. 46), o DP é:

[...] responsável por gerenciar o Backlog do Produto, garantir o retorno sobre o investimento, definir a visão do produto, gerenciar a entrada de novos requisitos e definir sua ordem, gerenciar o plano de releases, gerenciar o

orçamento e riscos do produto ou projeto e aceitar ou rejeitar o que será entregue ao final de cada iteração.

O DP também é o único que pode cancelar uma Sprint caso constate, em conjunto com a equipe de desenvolvimento, que essa Sprint já não faz mais sentido.

O Scrum Master detém o maior conhecimento sobre a metodologia, sendo responsável por auxiliar o DP na criação e na ordenação do Backlog do Produto, garantir que os princípios da metodologia estejam sendo cumpridos e que seus valores, principalmente, estejam sendo seguidos. Ele também tem como responsabilidade facilitar os eventos do Scrum e remover impedimentos enfrentados pela equipe de desenvolvimento.

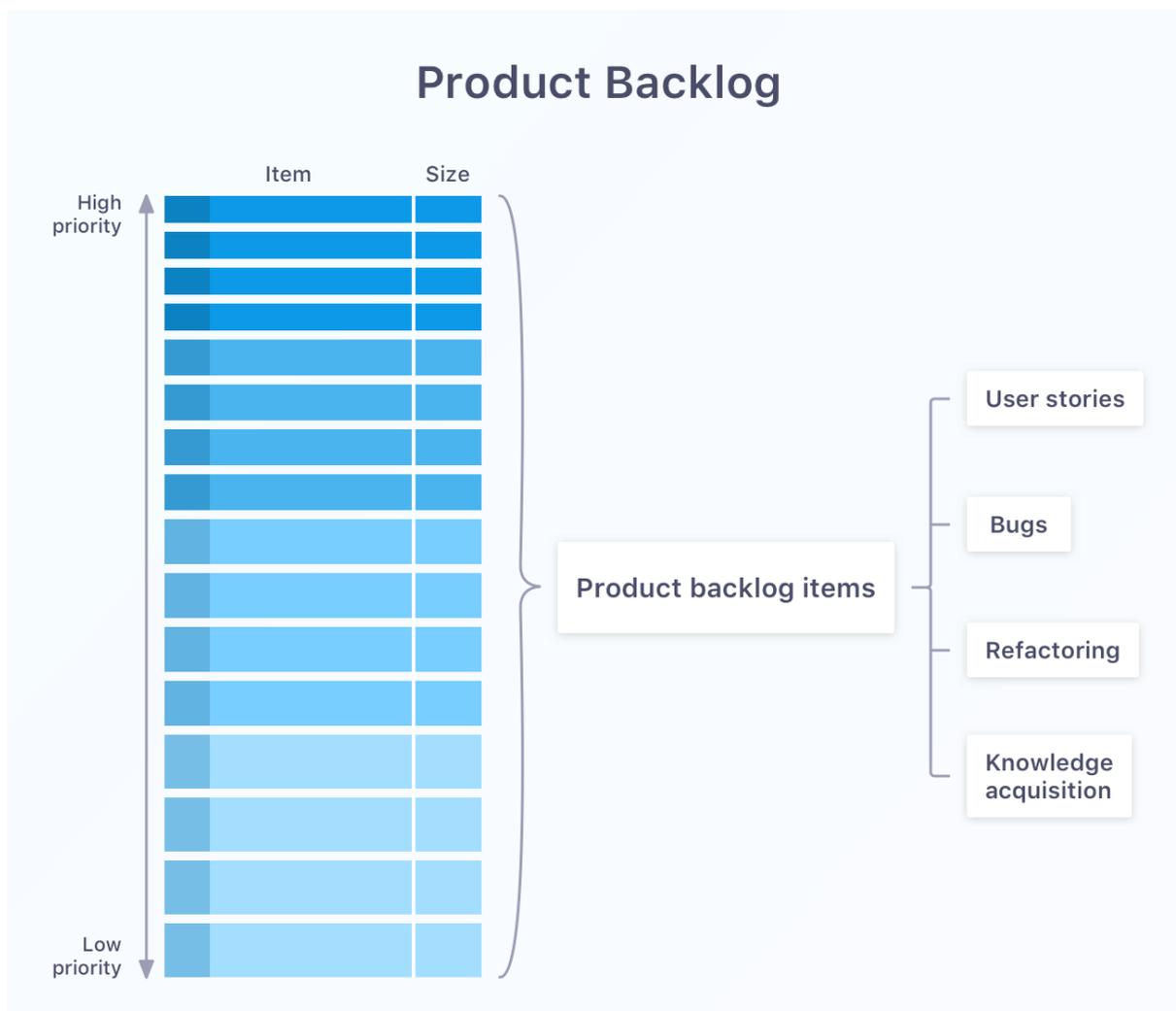
A equipe de desenvolvimento encarrega-se de desenvolver o produto, efetuar as entregas ao término de cada uma das interações e gerenciar que a entrega final do produto esteja atendendo a todos os requisitos informados no início do projeto e, sobretudo, os que surgirem durante o desenvolvimento, de forma que essa entrega seja feita com qualidade assim que o projeto for concluído.

2.2.3 Artefatos do Scrum

O Scrum possui alguns artefatos que são utilizados ao longo do projeto em diferentes momentos, a fim de organizar os entregáveis, mensurar as funcionalidades que estão por vir e as que estão sendo desenvolvidas.

O primeiro desses artefatos é o Backlog do Produto, que se caracteriza como uma lista ordenada criada pela equipe Scrum, e somente o DP pode inserir, excluir ou reordenar itens nessa lista. Organizada de forma crescente, os itens que a encabeçam aqueles com maior importância e que conseqüentemente serão desenvolvidos primeiro. Os últimos itens da lista são os menos importantes para o projeto ou precisam de um maior refinamento. A imagem a seguir (Figura 1) representa a ordenação dessa lista.

Figura 1 - Backlog



Fonte: Infinity (c2022)

O Backlog da Sprint é uma lista com todos os afazeres de determinada interação. O objetivo dessa etapa é garantir que, ao final da Sprint, todos os entregáveis que precisavam ser desenvolvidos sejam feitos da melhor forma possível. Ressalta-se que apenas os membros do setor de desenvolvimento podem inserir ou excluir tarefas nessa lista, visto que só eles têm conhecimento quanto ao andamento da Sprint (PRIKLADNICKI; WILLI; MILANI, 2014, p. 45).

Por fim, tem-se o incremento do produto, que se caracteriza por uma lista dos entregáveis desenvolvidos durante a Sprint e que objetiva demonstrar ao DP o que foi feito ao longo daquela interação. Devem ser entregues nessa etapa apenas os itens que foram finalizados, ou seja, que passaram por todas as etapas definidas.

2.2.4 Cerimônias do Scrum

Na metodologia Scrum, há alguns eventos fixos regulados em intervalos pré-programados, conhecidos como cerimônias do Scrum. Esses eventos são utilizados para revisão e planejamento dos passos desenvolvidos. A primeira dessas cerimônias é a Sprint, evento principal do Scrum. A Sprint é a forma utilizada pela metodologia para que a equipe de desenvolvimento possa trabalhar nas funcionalidades da aplicação que está sendo desenvolvida.

Cada Sprint tem duração de até um mês, o que permite feedbacks constantes do Dono do Produto quanto ao produto sendo desenvolvido. Da mesma forma, ele tem a possibilidade de analisar o que foi produzido e reorganizar o Backlog do Produto caso seja necessário (PRIKLADNICKI; WILLI; MILANI, 2014, p. 53).

A Sprint possui pequenos eventos que juntos a compõem, e o primeiro deles é a Reunião de Planejamento. Trata-se de uma reunião em que, no início do planejamento, a equipe se reúne a fim de planejar o que será feito naquela Sprint. A reunião é dividida em duas partes e tem como objetivo responder às seguintes perguntas: “O que será entregue no incremento resultante desta Sprint?” e “Como faremos para entregar o incremento nesta Sprint?” (PRIKLADNICKI; WILLI; MILANI, 2014, p. 54).

Para acompanhar o desenvolvimento do projeto, diariamente a equipe de desenvolvimento do produto se reúne para levantar os seguintes questionamentos: “O que fiz desde a última Scrum Diária?”, “O que pretendo fazer até a próxima Scrum Diária?” e “Existe algo me impedindo de concluir alguma tarefa?” (PRIKLADNICKI; WILLI; MILANI, 2014, p. 54). A partir dessa resposta, a equipe desenvolvedora decide se a sua estratégia de desenvolvimento está correta ou se precisa haver mudanças.

Ao final de cada Sprint, é feita a revisão da Sprint, ocasião em que os interessados pelo projeto se reúnem a fim de inspecionar o que foi trabalhado pela equipe de desenvolvimento. Nessa etapa, também se define o plano para a Sprint seguinte. Por fim, o último evento de uma Sprint é a retrospectiva, em que todos os membros da equipe Scrum se reúnem com o objetivo de aprimorar o processo para a próxima Sprint. Nessa etapa, também se identificam quais foram os problemas enfrentados nessa interação para que não se repitam nas próximas.

2.2.5 Vantagens do Scrum

Nos últimos anos, o número de empresas que adotam o Scrum vem aumentando de forma acelerada. Os relatos acerca da implantação trazem observações positivas sobre todo o processo:

[...] em 15 meses após adotar Scrum, 86% dos funcionários da Salesforce.com estão tendo uma ótima experiência ou a melhor experiência. Antes da adoção do Scrum, apenas 40% haviam dito isto. Na ePlan Services, após 9 meses utilizando o Scrum, a taxa de defeito por 1.000 linhas de código não comentadas caiu 70%. Na Yahoo!, 68% reportaram que o Scrum é melhor ou muito melhor do que a abordagem utilizada anteriormente. Além disso, a produtividade do time aumentou em média 36%, e 85% dos colaboradores disseram que eles continuariam a utilizar o Scrum se a decisão fosse apenas deles (PRIKLADNICKI; WILLI; MILANI, 2014, p. 56).

2.3 EXTREME PROGRAMMING (XP)

A XP é uma metodologia flexível que busca desenvolver projetos rapidamente para entregá-los dentro de um prazo predefinido mesmo que os requisitos sejam alterados durante o desenvolvimento do projeto. Essa metodologia utiliza o modelo incremental, ou seja, à medida que o software é utilizado pelo cliente, novas melhorias vão sendo incrementadas ao produto. Dessa forma, é possível garantir um produto que, ao final do projeto, vai gerar maior satisfação ao cliente e qualidade nas entregas.

A XP é uma metodologia de desenvolvimento de projetos de software com menos formalidade possível. Foi criada no final dos anos de 1990, mais precisamente em 1996, e apresentada por Kent Beck a OOPLSA (Object-Oriented Programming, Systems, Languages and Applications), em 2000, uma das maiores conferências de orientação a objetos reconhecida internacionalmente, que atualmente faz parte da System, Programming, Languages and Applications: Software for Humanity (SPLASH) (SBROCCO; MACEDO, 2012, p. 145).

A XP possui um conjunto de boas práticas a serem seguidas para que um projeto possa ser caracterizado como guiado por essa metodologia. A seguir, especificam-se as práticas que a XP preconiza e as suas principais características.

No início do projeto, é de suma importância que os desenvolvedores do produto se reúnam antes de começarem o desenvolvimento para adotar um padrão. Dessa forma, é possível que todos consigam entender mais rapidamente os códigos entre os membros da equipe. Além disso, cabe à equipe escolher a melhor linguagem de programação e as ferramentas que mais se ajustam aos requisitos do projeto.

Durante o desenvolvimento de um projeto guiado pela XP, busca-se o design simples a fim de evitar imprevistos, e os códigos são desenvolvidos por meio de uma implementação básica. Com um design simples, torna-se mais fácil conviver com alterações que o sistema possa vir a sofrer.

Procura-se, desta forma, evitar programações desnecessárias que não têm utilidade para o cliente em um primeiro momento. Na medida em que o sistema vai se estabilizando e adquirindo consistência em relação aos seus requisitos mais importantes, a equipe pode estudar a possibilidade de acrescentar essas “perfumarias”, caso necessário. A possibilidade de o sistema ser orientado a objetos facilita esse design, bem como as alterações solicitadas, mesmo que a linguagem de programação ou o framework sejam atualizados (SBROCCO; MACEDO, 2012, p. 149).

Diferentemente das tradicionais metodologias utilizadas no mercado, em que o cliente precisava muitas vezes adaptar os seus processos para a utilização do software, a XP preza pela qualidade do software entregue ao cliente. Dessa forma, sugere-se que o cliente acompanhe todo o processo de desenvolvimento e efetue os apontamentos necessários, a fim de que o produto o atenda da melhor forma possível – cliente sempre disponível ou presente.

À medida que o desenvolvimento do software avança, é imprescindível fazer pequenas entregas ao cliente, possibilitando que ele possa ter uma visão do software desenvolvido, sugerir mudanças e participar da criação dos testes conceituais, que serão os indicadores de uma boa codificação.

Quando a equipe recebe os requisitos do projeto, deve dividi-los em partes e encaminhá-los ao cliente. Dessa forma, o cliente efetua a classificação de cada um desses requisitos como prioridade alta, média ou baixa e, posteriormente, os organiza em ordem numérica – jogo de planejamento.

Após essa classificação, é de responsabilidade da equipe XP que retorne a prioridade desse projeto ao cliente com o custo e o tempo necessários para o desenvolvimento dessas funcionalidades. A equipe também é responsável por planejar as interações com o cliente, de modo que sejam feitas entregas frequentes a ele.

Assim como acontece no Scrum, na metodologia XP realiza-se diariamente uma reunião para alinhamento do projeto com tempo máximo de 15 minutos – stand up meeting. Aborda-se nessa reunião o que foi desenvolvido durante o dia anterior e o que será feito na data do dia vigente, assim como quais estão sendo os desafios e

os problemas encontrados. Essa reunião é feita com os colaboradores reunidos em pé para que se possa garantir que aconteça de fato em um curto espaço de tempo.

Segundo Sbrocco e Macedo (2012, p. 150), o conceito de *pair programming* também está presente em outras metodologias. Estudos apontam que programar em dupla pode render mais do que programar sozinho. Isso ocorre porque, na programação pareada, enquanto um desenvolvedor digita o código, o outro simultaneamente vai revisando, de forma a evitar erros de programação ou sugerir melhores estratégias de implementação. O desenvolvedor que digita é o condutor, e seu parceiro o navegador, ambos com o objetivo de resolver o mesmo problema em conjunto. A XP exige que todo código implementado seja feito em dupla e no mesmo computador.

Antes de efetuar as entregas constantes ao cliente, o desenvolvedor da funcionalidade a ser entregue deve revisar o seu código a fim de identificar se esse código está malfeito ou incorreto. Cabe ao desenvolvedor melhorá-lo, deixando-o o mais legível possível e sem erros – refactoring. Essa prática não visa refazer o código por completo, mas sim aperfeiçoar o que já foi feito tendo como base a entrega da melhor performance ao cliente.

Todo código gerado deve ser submetido a um teste automatizado, desde os testes mais básicos (unidade de código específica – para que seja possível verificar se os resultados gerados estão corretos) até os testes de aceitação junto ao cliente (para que seja possível validar se o software está atendendo ao que foi solicitado) – desenvolvimento guiado por testes.

A metodologia XP sugere que todos os desenvolvedores tenham acesso a todas as partes do código – código coletivo. Dessa forma, é possível que todos entendam a lógica por trás do código desenvolvido e, em caso de perda de algum membro da equipe durante a elaboração do projeto, é possível prosseguir com o desenvolvimento e com a manutenção naquela funcionalidade de código específica.

Metáforas facilitam a comunicação entre a equipe e os clientes por meio de analogias com os processos empregados. Tal artifício evita o constrangimento do cliente por não entender os termos técnicos usados pelos desenvolvedores. “A utilização de metáforas também colabora com a comunicação e o feedback, os quais [...] são valores indispensáveis da XP ligados ao relacionamento com o cliente” (SBROCCO; MACEDO, 2012, p. 151).

Em um projeto guiado pela metodologia XP, é proibido que a equipe cumpra horas extras. O ritmo sustentável sugerido pela XP não deve ultrapassar a carga horária de 40 (quarenta) horas semanais. Isso evita que o desenvolvedor perca a sua concentração devido ao cansaço e, conseqüentemente, cause falhas na implementação.

Seguindo os princípios da XP, devem ser feitas diariamente várias atualizações do produto para que os desenvolvedores e o cliente tenham sempre uma versão mais atualizada do software, melhorando assim o feedback – integração contínua. Indica-se que a equipe sempre valide a versão antes de atualizá-la para que sejam evitados bugs ou qualquer outro tipo de falha que possa ocorrer no produto. Essa integração é feita mediante um backup da versão imediatamente anterior para que o cliente não seja prejudicado caso ocorra uma falha no software.

A XP pretende com essa integração evitar o que pode acontecer nas metodologias mais tradicionais, em que o desenvolvedor só se preocupa com o que ele implementou e não sabe o que os outros estão fazendo em paralelo. A integração contínua proposta pela XP faz com que toda a equipe saiba das modificações ocorridas diariamente (SBROCCO; MACEDO, 2012, p. 152).

Assim como no Scrum, na XP efetuam-se as entregas ao cliente de forma gradativa – releases curtas. Dessa maneira, o cliente acompanha todo o processo de desenvolvimento, e a cada uma dessas etapas recebe uma parte do produto, diferentemente do que acontece com as metodologias tradicionais, em que o cliente recebe o produto por completo apenas no final do projeto.

2.3.1 Vantagens da XP

Como a XP é uma metodologia com muitos anos de mercado, utilizada por diversas empresas e com resultados comprovados, pode-se garantir que os processos propostos são de grande valia para as corporações, conforme explicitado abaixo:

A empresa Escrow adotou a Extreme Programming após inúmeros problemas em vários projetos, em que o custo e o prazo nunca eram precisos. Atua em projetos de e-commerce nos quais os requisitos mudam com muita frequência e necessitam de muita qualidade e velocidade para realizar transações on-line. Pela coleta de um pequeno conjunto de métricas simples, a Escrow foi capaz de medir o sucesso relativo do esforço de adoção da metodologia XP (SBROCCO; MACEDO, 2012, p. 158).

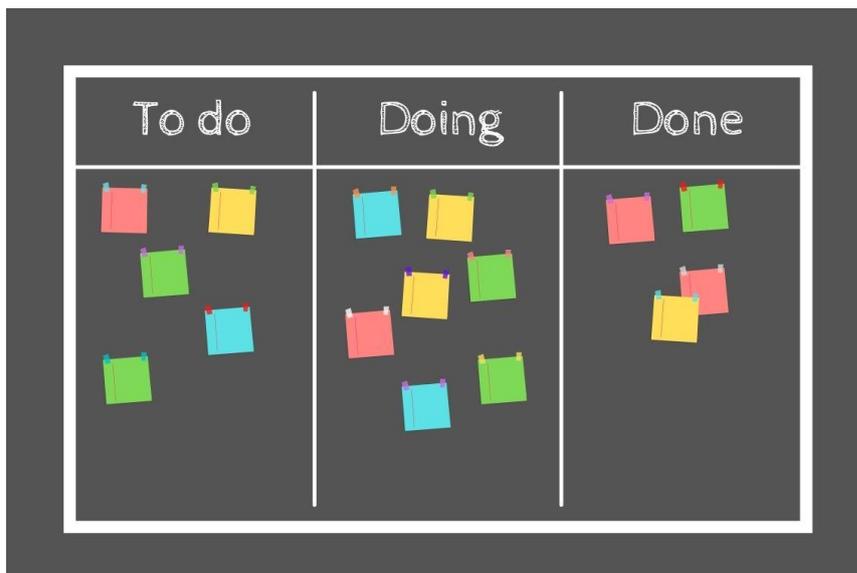
A empresa Primavera, que desenvolve projetos de alta complexidade, atendendo inúmeros clientes, e que possui um grupo de desenvolvedores das mais diversas linguagens e tecnologias, começou a utilizar a XP em seus projetos com o objetivo de levantar a autoestima de seus desenvolvedores, que sempre trabalhavam exaustivamente em projetos e não viam resultados. Muitas vezes, esses profissionais trabalhavam fora do horário habitual e ficavam descontentes com essa sobrecarga. “A XP resolveu esses problemas e o resultado da experiência é uma base de clientes altamente satisfeitos, e a equipe altamente motivada com um ambiente de desenvolvimento energético” (SBROCCO; MACEDO, 2012, p. 158).

2.4 KANBAN

Segundo Maschietto *et al.* (2021, p. 91), o Kanban, ou simplesmente cartão visual em japonês, é considerado um subsistema do TPS (Sistema Toyota de Produção), um mecanismo cujo objetivo é implementar um método de gerenciamento de projetos utilizando dois princípios: (1) visualizar o fluxo das atividades e (2) limitar o escopo do trabalho.

O Kanban possui um conceito simples em que alguns cartões são posicionados estrategicamente em uma tabela para que todas as tarefas e etapas de um projeto possam ser visualizadas com facilidade. Divide-se em três colunas – (1) to do, (2) doing e (3) done –, sendo as tarefas posicionadas na primeira das três colunas. A partir do momento em que as tarefas vão sendo desenvolvidas, os cartões são movidos na tabela, conforme mostra a imagem a seguir (Figura 2).

Figura 2 - Quadro Kanban



Fonte: Colaço (2021)

O funcionamento do Kanban no desenvolvimento de aplicações está centrado no quadro ilustrado na Figura 2, permitindo uma visualização de todo o ciclo de trabalho.

Um quadro típico do Kanban aplicado ao gerenciamento de projetos contém objetivos, listas de pendências, etapa de desenvolvimento, testes, urgência e tarefas concluídas.

De acordo com Maschietto *et al.* (2021, p. 92), “por ser uma metodologia flexível, cada departamento pode adaptar os quadros às suas necessidades, criando colunas para alocar cada uma das tarefas do projeto, assim como serão atribuídas aos membros da equipe”.

2.4.1 Vantagens do Kanban

A principal vantagem do Kanban é a rastreabilidade das tarefas, visto que ele possibilita o controle das atividades que estão sendo desenvolvidas por cada um dos membros da equipe.

O foco nas tarefas é o principal objetivo do Kanban; uma metodologia que impõe limites na execução de tarefas simultâneas – cada membro trabalha em um escopo reduzido para se concentrar na atividade e maximizar a qualidade do processo (MASCHIETTO *et al.*, 2021, p. 93).

2.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

Ao se trabalhar com desenvolvimento de softwares, é possível perceber que os requisitos levantados no início de um projeto dificilmente se manterão até o final sem algum tipo de alteração. O cliente final entende qual é a sua necessidade real apenas quando possui alguma funcionalidade do software em mãos. Dito isso, é muito raro que projetos que seguem metodologias tradicionais obtenham sucesso ao final de sua entrega, visto que, quando o cliente tem acesso à versão desenvolvida pela equipe após diversos meses, os requisitos raramente se mantêm os mesmos.

Além do fator citado acima, muitas vezes a entrega acaba sofrendo penalidades no quesito qualidade, pois como os requisitos são diferentes do solicitado pelo cliente, existe um forte retrabalho a ser feito no projeto. Com isso, pode-se concluir que projetos com entregas recorrentes e, conseqüentemente, com requisitos mutáveis têm um índice de sucesso maior quando comparados aos projetos cuja versão final é entregue apenas no término do projeto.

Segundo Sbrocco e Macedo (2012), num contexto imprevisível não se pode usar uma metodologia previsível. Dito isso, pode-se confirmar que o uso de uma metodologia ágil e que pregue o desapego à previsibilidade em desenvolvimento de software é indicado. Isso garantirá um sucesso maior ao projeto, fazendo com que os prazos possam ser cumpridos conforme o acordado com o cliente, apresentando qualidade, seguindo o orçamento estipulado e, principalmente, garantindo a satisfação do contratante, o que para o projeto é o requisito mais importante.

3 ASPECTOS METODOLÓGICOS

Esta pesquisa é caracterizada como qualitativa, pois tem como objetivo explicar um fenômeno ainda não conhecido em um determinado contexto.

A análise qualitativa é menos formal do que a análise quantitativa, pois nesta última seus passos podem ser definidos de maneira relativamente simples. A análise qualitativa depende de muitos fatores, tais como a natureza dos dados coletados, a extensão da amostra, os instrumentos de pesquisa e os pressupostos teóricos que nortearam a investigação. Pode-se, no entanto, definir esse processo como uma sequência de atividades que envolve a redução dos dados, a categorização desses dados, sua interpretação e a redação do relatório (GIL, 2017, p. 133).

Quanto aos procedimentos técnicos, a pesquisa enquadra-se como bibliográfica, tendo em vista que é elaborada com base em material já publicado (GIL, 2017). Como o intuito de criar uma metodologia para desenvolvimento de software para a empresa Beta a partir das propostas relacionadas, a metodologia ágil se situa como uma pesquisa aplicada, que tem como objetivo o desenvolvimento voltado a aplicações práticas, buscando solucionar um problema específico (GERHARDT; SILVEIRA, 2009).

Por fim, no tocante ao objetivo esta é uma pesquisa exploratória, pois tem como propósito proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses (GIL, 2017).

3.1 ETAPAS METODOLÓGICAS DE PESQUISA

Esta monografia tem como objetivo desenvolver uma metodologia que atenda, da melhor forma possível, aos padrões de projetos de desenvolvimento da empresa Beta. A fim de que esse objetivo seja alcançado, foram estabelecidas cinco etapas metodológicas para a execução do projeto, a saber: Etapa 1 - Fundamentação teórica; Etapa 2 - Modelo do processo atual da empresa; Etapa 3 - Identificação das fragilidades e possibilidades de melhoria do processo atual; Etapa 4 - Customização do processo a partir dos modelos ágeis; e Etapa 5 - Verificação do processo customizado. As etapas mencionadas são detalhadas nas seções que se seguem.

3.1.1 Fundamentação teórica

Na primeira etapa da monografia foram estudadas as metodologias ágeis emergentes a fim de subsidiar boas práticas e analisar a aderência dessas metodologias ao processo de desenvolvimento de software da empresa Beta. Com isso, foram detalhados padrões de três metodologias ágeis para encontrar pontos aderentes aos processos da empresa Beta e que possam ser implementados com o intuito de tornar o desenvolvimento de software mais ágil pelo setor de tecnologia da empresa.

3.1.2 Modelo do processo atual da empresa

Para dar conta da etapa de customização do processo de desenvolvimento de software de forma qualificada e eficiente, é necessário o entendimento do processo atual da empresa Beta. Para tanto, esta etapa mapeou o processo atual da empresa por meio de um diagrama de processos identificando atores, responsabilidades e atividades.

3.1.3 Identificação das fragilidades e possibilidades de melhoria do processo atual

Na terceira etapa identificaram-se objetivamente num primeiro momento as fragilidades do processo atual a partir de uma análise crítica do processo e da experiência do pesquisador nesse processo. Em um segundo momento foram apontadas possíveis melhorias para qualificação de um novo processo.

3.1.4 Customização do processo a partir dos modelos ágeis

Para que seja possível atingir um nível de excelência no processo de desenvolvimento de software na empresa Beta, nesta etapa metodológica realizou-se a customização da metodologia da empresa utilizando-se como base métodos, técnicas identificadas durante a etapa de fundamentação teórica e experiência dos pesquisadores.

3.1.5 Verificação do processo customizado

A verificação do processo teve como objeto o impacto da customização, a pertinência das propostas e a aderência às necessidades entendidas como prioritárias pela empresa. Nesta etapa, avaliou-se se o novo processo é aderente aos processos da empresa Beta e ao seu atual modelo de negócio.

4 DESENVOLVIMENTO

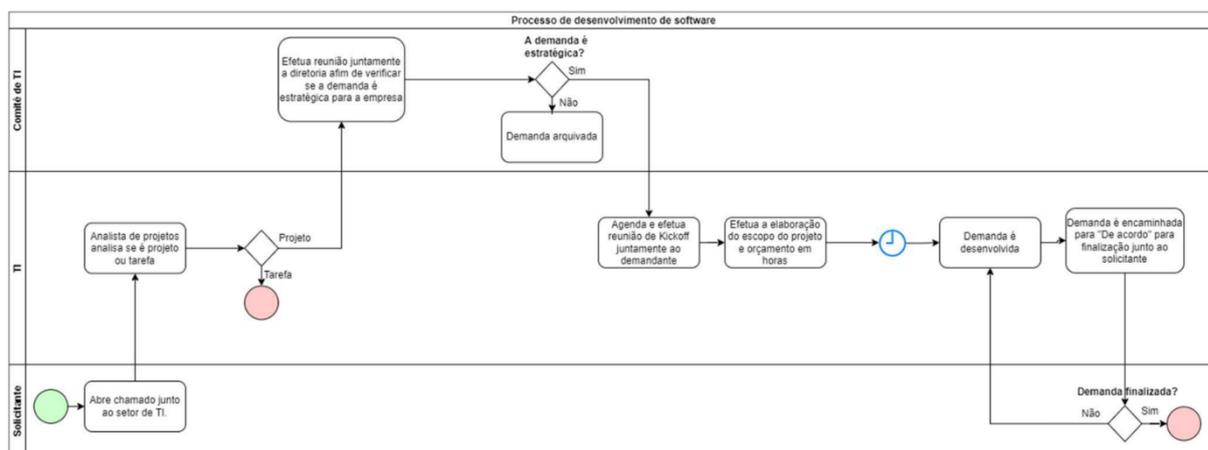
4.1 MODELO DO PROCESSO ATUAL DA EMPRESA BETA (AS-IS)

Atualmente o processo utilizado pela empresa Beta no desenvolvimento dos projetos possui fragilidades que tornam esse desenvolvimento não ágil. Entre essas fragilidades estão as entregas realizadas apenas na finalização do projeto. Os requisitos do projeto são levantados durante a reunião de kick off – realizada na fase inicial de um projeto com o intuito de alinhar todos os detalhes entre os envolvidos – e passam pela diretoria, que decide se o projeto é estratégico e se terá continuidade ou não. Esse processo de decisão pode levar dias, ou até mesmo semanas, o que faz com que haja uma mudança de requisitos já no início do projeto.

Durante o desenvolvimento da aplicação, muitas vezes essa mudança de requisitos também ocorre, tendo em vista que as entregas são feitas apenas no final do projeto.

O diagrama “Processo de desenvolvimento de projeto”, ilustrado na Figura 3, apresenta o processo atual desde o momento em que surge a demanda até a entrega do produto.

Figura 3 - Processo de desenvolvimento de projeto



Fonte: Elaboração do autor (2022)

Os setores que compõem o comitê de TI da empresa Beta têm o objetivo de avaliar os projetos em aberto e de classificá-los quanto à sua importância. Listam-se a seguir os setores e as suas respectivas responsabilidades.

- Diretoria – tem por objetivo definir se a atividade deve ou não ir para a fila de desenvolvimento e classificar a sua prioridade.
- Setor de Tecnologia – a responsabilidade do setor de TI é executar de forma devida as demandas propostas pelo solicitante.
- Setor Demandante – responsável por abrir a demanda e acompanhá-la do início até o momento da entrega.

Atualmente na empresa Beta o processo funciona de forma sequencial: o demandante abre o projeto por meio de um formulário; posteriormente, o analista de projetos avalia se esse projeto é procedente ou não; se for procedente, efetua-se uma reunião de kick off, e a demanda é levada à diretoria; caso a diretoria aprove o projeto, a demanda vai para a fila de desenvolvimento e o projeto é desenvolvido; por fim, esse projeto é colocado em homologação e, se o cliente o aceitar, o projeto é finalizado.

Além das vulnerabilidades identificadas no processo, percebe-se que a visualização das tarefas pelos desenvolvedores é complexa. Hoje em dia na empresa essa visualização é feita em formato tabular, o que dificulta a mensuração da quantidade de atividades e responsabilidades de cada membro da equipe. Segue abaixo um exemplo de tela de processo de desenvolvimento de projeto:

Figura 4 - Tela de processo para desenvolvimento de projeto

▼ Projetos em desenvolvimento							
Elemento	Respo...	Dev tempo	Status	Complexidade	Setor demandante	Timeline	
Projeto de desenvolvimento 1		4h 54m 52s	Fazendo	Avançado	Logística	abr 25 - jun 10	
Projeto de desenvolvimento 2			A Fazer	Alto	Suprimentos	jun 10 - 24	
Projeto de desenvolvimento 3			A Fazer	Médio	Suprimentos	jun 24 - jul 11	
Projeto de desenvolvimento 4			A Fazer	Médio	Controladoria	jul 18 - 30	
Projeto de desenvolvimento 5			A Fazer	Avançado	Qualidade	jul 30 - ago 6	
Projeto de desenvolvimento 6			A Fazer	Avançado	Controladoria	ago 8 - set 1	
+ Adicionar Elemento		4h 54m					
▼ Tarefas em desenvolvimento							
Elemento	Respo...	Dev tempo	Status	Complexidade	Setor demandante	Timeline	
Projeto de desenvolvimento 7		16m 0s	Aguardando terce...	Médio	Marketing	jun 13 - 24	
Projeto de desenvolvimento 8			Fazendo	Médio	TI	mai 12 - jul 30	
Projeto de desenvolvimento 9			Fazendo	Avançado	Diretoria	mai 17 - jun 30	
Projeto de desenvolvimento 10			Fazendo	Alto	TI	jun 27	
Projeto de desenvolvimento 11			Aguardando terce...	Alto	Controladoria	mai 3 - jun 16	
Projeto de desenvolvimento 12			A Fazer	Alto	Suprimentos	jun 14 - 25	
Projeto de desenvolvimento 13			A Fazer	Baixo	Qualidade	jun 10	
Projeto de desenvolvimento 14		2h 34m 0s	Aguardando terce...	Médio	Qualidade	jun 10	

Fonte: Elaboração do autor (2022)

A tela é composta por dados relacionados à demanda proposta para desenvolvimento, sendo estes o título da demanda, o desenvolvedor responsável, o tempo dedicado à demanda, o status do desenvolvimento, a complexidade do projeto, o setor demandante e as datas de início e de fim.

Detalhando-se o conteúdo das colunas, na coluna Status constam os seguintes passos: Finalizado, Fazendo, A fazer, Parado, Aguardando Terceiro, Homologação e Cancelado. Na coluna Complexidade, a informação é “calculada” em horas, sendo de baixa complexidade projetos entre 5 e 10 dias, de média complexidade projetos entre 10 e 30 dias, de alta complexidade projetos entre 30 e 60 dias e avançados os projetos superiores a 60 dias. Por fim, a timeline corresponde ao tempo de duração do projeto em dias corridos.

4.2 SUGESTÕES DE MELHORIA DO PROCESSO ATUAL

A sugestão do uso de metodologias ágeis no processo de desenvolvimento de software vai ao encontro da tendência do mercado de ser cada vez mais competitivo e, principalmente, de buscar mais qualidade nas entregas dos projetos. Dito isso, a implementação das metodologias ágeis no estudo de caso da empresa Beta tem como objetivo assegurar ao cliente maior satisfação no final do projeto e dar mais liberdade de atuação aos desenvolvedores na condução desse projeto.

Ao se debruçar sobre o problema, o pesquisador considera que a implementação de um modelo baseado em metodologias ágeis híbridas – Scrum, Kanban e XP – é uma aposta que pode trazer ao processo produtivo atual da empresa Beta os requisitos de eficiência e agilidade necessários para promover a construção de um fluxo ágil e com entregas de maior recorrência.

4.2.1 Conceitos do Scrum e visão pontual dos processos da empresa Beta

Na sequência serão descritas, a partir dos conceitos propostos pelo Scrum, as possibilidades de inovações e melhorias do processo atual da empresa Beta.

4.2.1.1 Reunião de planejamento da Sprint

Inserir no processo a reunião de planejamento da Sprint propõe que o time esteja alinhado com o Product Owner (gestor de TI) sobre o que deve ser priorizado a fim de gerar valor para o cliente de forma rápida e eficiente. Com essa priorização alinhada com o time de desenvolvimento, esses profissionais vão gerar estimativas e detalhar o que será desenvolvido nas tarefas (em um nível mais técnico). Tal ação deve simplificar o desenvolvimento dessas tarefas. A reunião deverá ocorrer no início de cada mês a fim de que seja planejado o que será desenvolvido na Sprint de um mês.

4.2.1.2 Daily Sprint

A Daily Sprint traz o conceito de que o time esteja alinhado no que diz respeito ao que cada um está trabalhando, mas principalmente para que o planejamento da Sprint não se perca. Esse rito acontecerá todos os dias no primeiro horário da manhã, momento em que o Scrum Master (analista de projetos de TI) se reunirá com o time para uma conversa breve (de aproximadamente 15 minutos). Nesse rito, o time deve seguir um roteiro com as seguintes perguntas: “O que eu fiz desde a última Daily?”, “O que eu vou fazer depois dessa Daily?” e “Tem algo impedindo o meu trabalho?”. Com as respostas dessas perguntas, o Scrum Master poderá acompanhar o time no desenvolvimento de projetos e no andamento das tarefas. Sendo ativo e prospectivo, deve reportar ao Product Owner quaisquer tipos de empecilho que possam dificultar o andamento dos projetos.

4.2.1.3 Revisão da Sprint

A revisão da Sprint terá como objetivo efetuar a entrega feita na última Sprint aos *stakeholders*, fazendo dessa forma com que as entregas deixem de ser efetuadas apenas no final do projeto. Isso permite a realização das entregas de forma gradativa, podendo o demandante acompanhar o desenvolvimento do projeto na íntegra. Não haverá um rito de como essa entrega deve ser feita, se apenas uma apresentação informal para o cliente mostrando a funcionalidade em um notebook ou se uma

apresentação mais formal, contudo é indispensável que sejam apresentadas sempre entregas que gerem valor ao demandante.

4.2.2 Conceitos da XP

Para o desenvolvimento deste trabalho, o pesquisador optou pela utilização da metodologia XP, visto que ela aborda em alguns de seus pontos dores latentes encontradas na companhia Beta, como o não envolvimento dos usuários na etapa de testes e, principalmente, a não refatoração dos códigos. Códigos antigos utilizados pela empresa, não performáticos e com um baixo nível de maturidade acabam por trazer inúmeros problemas para a equipe.

4.2.2.1 Testes de usuários

Nessa etapa, antes do time de desenvolvimento levar a funcionalidade desenvolvida para a revisão da Sprint e de seguir o fluxo para a produção, essa funcionalidade deve ser disponibilizada ao usuário para testes. É importante que o usuário possa testar a funcionalidade com pelo menos uma semana de antecedência. Assim, evitam-se retrabalhos na próxima Sprint, tendo em vista que o usuário testará a funcionalidade antes de ser entregue em produção.

4.2.2.2 Código coletivo

Atualmente o código desenvolvido pelos desenvolvedores é de conhecimento individual, apenas o autor do desenvolvimento conhece o algoritmo. Com a inserção do código coletivo, deve-se criar um ambiente de desenvolvimento em que todos do time tenham acesso ao código desenvolvido com cada um dos Devs (desenvolvedores da equipe).

4.2.2.3 Refactoring

Nessa rotina o time de desenvolvimento, pelo menos uma vez por mês, revisará o código desenvolvido nas últimas Sprints a fim de propor melhorias, torná-lo mais limpo (bem indentado, com pouca repetitividade, poucos comentários, tratamento nos

erros, etc.) e legível, e principalmente propiciar uma melhor compreensão da lógica envolvida.

4.2.2.4 Programação em pares

Antes de qualquer funcionalidade ser lançada em produção (previamente à revisão de Sprint), é de responsabilidade do desenvolvedor responsável pelo código apresentá-lo a um outro desenvolvedor a fim de garantir a eficiência e a transparência do código, gerando dessa forma uma espécie de adaptação na programação em pares.

Estudos apontam que programar em dupla pode render mais do que programar sozinho. Isso ocorre porque, na programação pareada, enquanto um desenvolvedor digita o código, o outro simultaneamente vai revisando, de forma a evitar erros de programação ou sugerir melhores estratégias de implementação (SBROCCO; MACEDO, 2012, p. 150).

Dessa forma, garante-se um código com um nível cada vez mais alto e o desenvolvimento dos profissionais do time de TI.

4.2.3 Conceitos do Kanban

No ponto a seguir, a ideia proposta pelo pesquisador é elaborar um modelo de cartões (Kanban) para os desenvolvedores da empresa Beta. A principal vantagem desse modelo de cartões será a visualização em um formato mais limpo e fluido, diferentemente do modelo de lista utilizado na atualidade pelo time de desenvolvedores.

Nos cartões, constarão as seguintes etapas: A Fazer, Fazendo, Parado, Aguardando Terceirizado, Homologação e Finalizado. Nesse modelo, os desenvolvedores poderão controlar as suas tarefas de forma mais objetiva, apenas visualizando as tarefas da Sprint atual, diferentemente de como é hoje (formato de lista), em que se tem acesso também às tarefas das próximas Sprints.

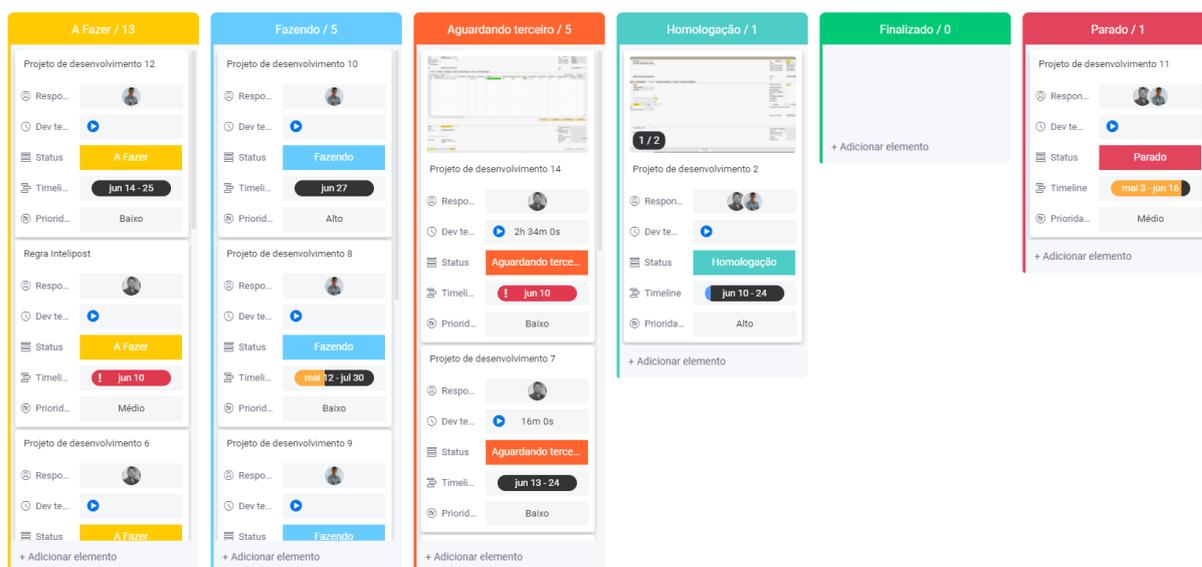
Os cartões apresentam algumas informações-chave às quais os Devs terão acesso. Essas informações são: Título da tarefa – identifica a demanda; Responsável – define o desenvolvedor responsável pela tarefa/projeto; Descrição da tarefa – descreve o briefing da demanda; Timeline – define o prazo de desenvolvimento e a

prioridade que identificará a importância daquela tarefa, conforme estabelecido pelo Scrum Master.

Essa ação vai permitir a visualização apenas dos campos-chave necessários ao desenvolvimento da funcionalidade proposta.

A Figura 5 a seguir ilustra um modelo de proposta de visualização.

Figura 5 - Tela de Kanban para desenvolvimento de projeto



Fonte: Elaboração do autor (2022)

5 VERIFICAÇÃO DO PROCESSO (TO-BE)

A etapa de verificação tem como objetivo avaliar, apoiando-se em algumas perguntas-chave, a aderência do processo baseado em metodologias ágeis proposto ao time de tecnologia da empresa Beta. Para isso, foram entrevistados três membros da empresa que possuem cargos gerenciais ou que participam ativamente do processo atual. A abordagem To-Be propõe uma visão dos processos futuros de uma organização a partir da implementação de melhorias.

Para o desenvolvimento da entrevista, foi feita uma apresentação aos entrevistados com o objetivo de contextualizar a inserção do modelo ágil no processo atual e assim apontar quais seriam os benefícios adquiridos. Após a apresentação, o entrevistador questionou cada um dos participantes com as perguntas abaixo:

- Qual a sua opinião quanto ao processo proposto?
- Você tem sugestões de melhoria?
- Você considera que o novo processo está adequado à estratégia corporativa da empresa?
- Você considera que esse processo contribui com a cadeia de valor agregado da empresa?

5.1 RESULTADOS DA VERIFICAÇÃO DO PROCESSO (TO-BE)

Para avaliar o processo proposto, o entrevistador realizou reuniões de no máximo uma hora com cada um dos candidatos separadamente para apresentação do processo proposto e posterior coleta de respostas. As reuniões foram feitas no modo presencial, e o entrevistador coletou as respostas com um formulário após a apresentação. Durante a coleta das informações, o entrevistador ponderou qualquer tipo de dúvida dos entrevistados.

5.1.1 Primeiro entrevistado

O participante 1 é desenvolvedor sênior na empresa Beta há aproximadamente um ano e meio. Possui 36 anos de idade e atua há duas décadas no mercado de

tecnologia. Indagado sobre a proposta apresentada e sobre as modificações do processo, esse participante comentou:

[...] agregará muito ao processo atual da empresa, tendo em vista que preenche lacunas e problemas latentes que possuímos hoje, como, por exemplo, a mudança de requisitos constante que temos hoje com os demandantes (Participante 1).

Acerca das sugestões de melhoria, esse participante acredita que o maior impacto seria reduzir a Sprint para uma semana, diferentemente do proposto no modelo. Isso garantiria mais entregas e revisões de Sprint e, por consequência, maior qualidade. As propostas relacionadas ao Kanban permitirão uma visão completa de todos os projetos, e não apenas das atividades a serem feitas na Sprint. Quando questionado sobre a adequação do novo processo à estratégia corporativa da empresa, o participante 1 respondeu: “Sim, esse processo vai ao encontro do planejamento estratégico da empresa”.

Na questão sobre a contribuição do modelo ágil e sua relevância para a cadeia de valor agregado da empresa, o participante 1 enalteceu o uso desse modelo: “Sim, pois atualmente nossos processos estão defasados, usando modelos tradicionais como de cascata. Acredito que aplicando essa metodologia, seguiremos de uma forma mais ágil”.

5.1.2 Segundo entrevistado

O participante de número 2 é o gerente de TI da empresa Beta. Ele possui 47 anos de idade e trabalha no mercado de tecnologia há aproximadamente 22 anos. No que se refere à proposta, o entrevistado 2 pontuou a seguinte opinião: “Acredito que seja uma excelente ideia, [...] pois vejo que essa é uma ótima proposta para simplificar o processo atual e deixar mais transparente com o solicitante a forma com que a TI executa seus projetos”.

Quanto à sugestão de melhoria, o entrevistado apontou que em sua opinião deveria ser feito um tipo de “trabalho” com o time de TI atual da empresa Beta, tendo em vista que eles não estão acostumados a trabalhar com metodologias ágeis: “Deve-se incluir na proposta uma forma de buscar o engajamento dos usuários nessa nova tratativa de modelos ágeis” (Participante 2).

No que se refere à aderência do processo proposto à estratégia corporativa da empresa Beta, o entrevistado alegou que esse processo vai ao encontro da visão da empresa e do crescimento almejado, motivo pelo qual é preciso desenvolver uma gestão de projetos de TI adequada às tendências atuais do mercado. Por fim, ao ser questionado sobre a contribuição do processo com a cadeia de valor da empresa, o entrevistado afirmou que a adoção da mentalidade ágil por parte da TI pode fomentar uma propagação do movimento por toda a empresa, e dessa forma o processo pode colaborar massivamente quanto ao valor agregado.

5.1.3 Terceiro entrevistado

O participante de número 3 é o diretor de operações da empresa Beta. Ele possui 44 anos e trabalha há aproximadamente 27 anos como gestor/diretor de operações. Ao ser questionado quanto ao processo proposto, o entrevistado concluiu:

Minha opinião é que a proposta está aderente ao nível de maturidade dos processos de TI atuais, bem como ao nível de comunicação desenvolvido com as demais áreas da empresa até o momento. No entanto, é importante garantir que a equipe interna consiga se adaptar a esse modelo, sobretudo o time de desenvolvimento, pois já tentamos algo parecido no passado mas o time não se adaptou (Participante 3).

Ao ser questionado quanto à oportunidade de melhoria, ele apontou que, a fim de garantir que o processo seja eficiente, além da proposta deve-se realizar um acompanhamento nos indicadores de performance de TI para avaliar o impacto desse modelo nos indicadores.

No que tange à adequação do processo à estratégia corporativa de empresa, o entrevistado concluiu que o processo está adequado, pois a implementação de modelos ágeis se comunica com a situação de maturidade da gestão da empresa. Ele ainda afirmou que para a escalabilidade do negócio será necessário trabalhar de forma ágil, pois já se nota uma maior complexidade nos processos desenvolvidos pela TI.

Por fim, quanto à contribuição do processo com a cadeia de valor, o entrevistado concluiu:

Sim, o processo está adequado, pois a empresa Beta realizou recentemente uma revisão da sua cadeia de valor, reposicionando a TI como processo

estratégico essencial para suportar as ambições de crescimento da companhia. A implementação de métodos ágeis dará celeridade às entregas e às percepções de performance e satisfação dos clientes internos, trazendo impacto direto na valorização e na satisfação das demais partes interessadas (clientes externos, parceiros, governo e sociedade) (Participante 3).

5.2 CONCLUSÕES DAS ENTREVISTAS

A partir da análise dos apontamentos feitos pelos entrevistados, pôde-se concluir que as sugestões de melhoria são aderentes às estratégias e aos processos da empresa Beta. Uma vez efetivadas as correções apontadas, deve ser iniciada a implantação do processo proposto. O acompanhamento contínuo do novo processo será responsabilidade do pesquisador, de modo que seja garantida a aderência desse novo processo aos processos atuais do time de TI da empresa Beta. Assim, garante-se que o novo processo potencializará o nível da qualidade das entregas e, principalmente, proporcionará maior celeridade ao desenvolvimento de software da empresa.

6 CONCLUSÃO E TRABALHOS FUTUROS

O principal objetivo desta pesquisa foi o desenvolvimento de uma metodologia ágil customizada para a empresa Beta, com o olhar atento aos processos já existentes e às peculiaridades presentes nessa empresa. Conforme o estudo, o desenvolvimento e o conhecimento gerado a partir de diversas metodologias, um híbrido de três metodologias foi gerado a fim de atender à proposta estabelecida no objetivo geral desta monografia.

Para cumprir os objetivos propostos, foi necessária a imersão do pesquisador no processo atual da empresa. A imersão no dia a dia da empresa para compreender como o processo realmente acontece e se efetiva pela equipe apontou peculiaridades desconhecidas ao pesquisador. Assim se enriqueceu o conhecimento acerca dos processos organizacionais e da própria equipe e se aquilatou a importância da qualificação do processo para uma melhor produtividade na entrega. A metodologia ágil foi entendida pelo pesquisador como um rico recurso no aprimoramento do processo atual da empresa, evidenciando a possibilidade de melhoria em diferentes aspectos, tais como a comunicação da equipe, a gestão do processo, a colaboração do time, entre outros. Ao estudar os artefatos e processos propostos e referenciados na revisão bibliográfica, o pesquisador colocou em prática, de forma significativa, a construção de um conhecimento que está tendo a sua experimentação em execução. A prática desse conhecimento permite o refinamento de conceitos e a reflexão crítica de novas oportunidades para a qualificação do processo da empresa.

Na análise dos dados relacionados à verificação do novo processo, entende-se que se alcançaram objetivos importantes propostos na monografia. A adequação do processo, conforme os apontamentos efetuados pelos entrevistados da empresa Beta, deve ser realizada como trabalhos futuros, promovendo assim a melhoria efetiva do processo da empresa, a implantação e o acompanhamento contínuo.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, Gustavo. Principais metodologias ágeis e as vantagens para os negócios. **Digilandia**, 17 maio 2020. Seção Gestão e Liderança. Disponível em: https://digilandia.io/gestao-e-lideranca/metodologias-ageis/#Vantagens_das_metodologias_ageis. Acesso em: 24 set. 2021.

COLAÇO, Janize. Método Kanban: saiba como ele pode ajudar na organização e fluidez de tarefas. **Na prática.org**. 1 jun. 2021. Disponível em: <https://www.napratica.org.br/metodo-kanban/>. Acesso em: 10 fev. 2022.

FUNDAÇÃO INSTITUTO DE ADMINISTRAÇÃO. **Gestão de projetos**: o que é, importância e PMI. 22 out. 2018. Disponível em: <https://fia.com.br/blog/gestao-de-projetos/>. Acesso em: 24 set. 2021.

GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo (org.). **Métodos de pesquisa**. Porto Alegre: Editora da UFRGS, 2009. Disponível em: <http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>. Acesso em: 10 jun. 2022.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 6. ed. Grupo GEN, 2017. 9788597012934. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788597012934>. Acesso em: 22 nov. 2021.

INFINITY. **Product backlog**. c2022. Site. Disponível em: <https://startinfinity.com/product-management-framework/product-backlog>. Acesso em: 15 abr. 2022.

LOSNAK, Giulia. Metodologia ágil: o que é? **Alura**, 12 jun. 2019. Disponível em: https://www.alura.com.br/artigos/o-que-e-metodologia-agil?gclid=Cj0KCQjwv5uKBhD6ARIsAGv9a-zM15w5Sy9sEQ7jSp_jb2MYEB2yb-o23lSnh4Fno0uUetgB7hgQQQsaAIJTEALw_wcB. Acesso em: 19 set. 2021.

MASCHIETTO, Luís Gustavo *et al.* **Desenvolvimento de software com metodologias ágeis**. Grupo A, 2021. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556901824/>. Acesso em: 27 out. 2021.

NOLETO, Cairo. Manifesto Ágil: saiba como surgiu e quais os 12 princípios. **Trybe**, 1 maio 2020. Seção Carreira. Disponível em: <https://blog.betrybe.com/carreira/manifesto-agil/>. Acesso em: 19 set. 2021.

PRESSMAN, Roger S. **Engenharia de software**. São Paulo: Pearson Education do Brasil, 1995. 1055 p.

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos ágeis para desenvolvimento de software**. Grupo A, 2014. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582602089/>. Acesso em: 27 out. 2021.

SANCHES, Rosely; BARBOSA, Ellen F. **Qualidade de software**: avaliação de sistemas computacionais. Apresentação em Power Point. Disponível em: https://edisciplinas.usp.br/pluginfile.php/57546/mod_resource/content/1/Aula8-QualidadeSoftware.pdf. Acesso em: 24 set. 2021. Apresentação PPT.

SARTO, Paula del. Metodologias ágeis: quais são, como funcionam e como se capacitar para os tipos mais demandados pelas empresas de tecnologia? **Impulso Network**, 30 nov. 2018. Disponível em: https://blog.impulso.network/metodologias-ageis/?utm_source=google&utm_medium=cpc&utm_campaign=ad-search-blog-impulso&gclid=Cj0KCQjwv5uKBhD6ARIsAGv9a-zYhHkZaINFjH7yte9bx-wzTa7JKpczVo9qTlgS2uVQS-ykSOFZXscaAkQiEALw_wcB. Acesso em: 19 set. 2021.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de Macedo. **Metodologias ágeis**: engenharia de software sob medida. São Paulo: Érica, 2012.

WARD CUNNINGHAM. **Manifesto for Agile Software Development**. c2001. Disponível em: <http://agilemanifesto.org>. Acesso em: 20 mar. 2022.