



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

UNIDADE ACADÊMICA DE SERRA TALHADA

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GABRIEL MATHEUS DOS SANTOS

Implementação de Metodologias Ágeis por Ferramentas de Gerenciamento de Projeto

Serra Talhada,
2023

Gabriel Matheus dos Santos

Implementação de Metodologias Ágeis por Ferramentas de Gerenciamento de Projeto

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Unidade Acadêmica de Serra Talhada da Universidade Federal Rural de Pernambuco como requisito parcial à obtenção do grau de Bacharel.

Orientador: Prof. Dr. Ítalo Cesar de Souza Belo
Coorientador: Prof. Dr. Paulo Mello da Silva

Serra Talhada,
2023

Implementação de Metodologias Ágeis por Ferramentas de Gerenciamento de Projeto

Gabriel Matheus dos Santos¹, Ítalo Cesar de S. Belo¹, Paulo Mello da Silva¹

¹Universidade Federal Rural de Pernambuco (UFRPE)
Caixa Postal 063 – 56.900-000 – Serra Talhada – PE – Brasil

{gabrielmatheus2503,paulomellosilva2}@gmail.br, italobelo@yahoo.com.br

Abstract. *With the emergence of agile methods, software project management began to be treated differently from traditional management. Using an agile approach to software development, several tools have been and are being developed to assist project managers. This makes it difficult for project managers to choose tools. Given the growing adoption of project management tools, the use of mobile applications, and the difficulties related to selecting a tool present in the market, this article will indicate how project management tools implement agile methods. Therefore, a study will be carried out on agile methodologies and project management tools most used in the market, as well as verifying how much two most used tools currently implement Scrum and Kanban methodologies.*

Resumo. *Com o surgimento dos métodos ágeis, o gerenciamento de projeto de software passou a ser tratado de forma diferente da gestão tradicional. Utilizando abordagem ágil nos desenvolvimento dos software, várias ferramentas foram e estão sendo desenvolvidas para auxiliar os gerentes de projetos. Isso dificulta a escolha de ferramentas pelos gerentes de projetos. Dada a crescente adoção das ferramentas de gerenciamento de projeto, o uso dos aplicativos móveis e as dificuldades relacionadas à escolha de alguma ferramenta presente no mercado, este artigo, indicará como as ferramentas de gerenciamento de projeto implementam os métodos ágeis. Portanto, será feito um estudo sobre metodologias ágeis e ferramentas de gerenciamento de projeto mais utilizadas no mercado, bem como a verificação do quanto duas ferramentas mais utilizadas atualmente implementam as metodologias Scrum e Kanban.*

1. Introdução

O processo de desenvolvimento de software consiste no conjunto de atividades executadas com o objetivo de gerar qualquer produto de software. Nesse contexto, o gerenciamento de criação de um software envolve desde a análise de viabilidade para criação do sistema até a implantação do mesmo. Isso é realizado através da gestão de projetos de software [Sommerville 2011, Pressman and Maxim 2016].

A gestão de projetos de software envolve desde as atividades de planejamento, organização, supervisão até o controle sobre aspectos relacionados ao projeto de criação de um produto de software [Sommerville 2011, Carvalho 2019]. Ela abrange o gerenciamento de escopo, do tempo, de custo e da qualidade [Cavalcanti 2016], fazendo a integração entre conhecimento, habilidade, técnicas e ferramentas para atender os requisitos do projeto [PMI 2017].

Para satisfazer as necessidades das partes interessadas (*stakeholders*), ou superá-las, a gestão de projetos de software auxilia na identificação das necessidades dos mesmos, na

definição clara de objetivos alcançáveis além de conduzir o balanceamento dos objetivos do projeto em diferentes prioridades, de acordo com as expectativas dos *stakeholders* [Cavalcanti 2016]. Nesse cenário, são empregadas em diversas empresas diferentes metodologias ágeis na gestão de projetos.

As metodologias ágeis correspondem a uma abordagem diferente da gestão de projetos tradicionais definidas no PMBoK [Carvalho 2019]. Elas são baseadas no manifesto ágil, que surgiu em 2001, a partir de uma reunião de líderes de desenvolvimento de softwares [Sutherland 2014], que focou na resolução dos seguintes problemas do desenvolvimento de softwares [Beck et al. 2001]; muito tempo gasto no processo de documentação e não no desenvolvimento da solução [Sommerville 2011, Drumond 2021], mudanças durante a execução do projeto gerando custo muito elevado e retrabalho excessivo. Como resultado o software final é entregue muito tempo depois do prazo que foi estipulado com o cliente [Sommerville 2011, Pressman and Maxim 2016, Drumond 2021].

O Scrum é um *framework* que entrega valor e qualidade por meio de soluções adaptativas, permitindo a incorporação de práticas de outros métodos ágeis para resolver problemas complexos, otimizando o trabalho de pessoas, equipes ou de uma empresa [Schwaber and Sutherland 2020]. Já que não é voltada apenas para desenvolvimento de software, mas para qualquer setor, essa abordagem vem sendo empregada por algumas das maiores empresas do mundo [Schwaber and Sutherland 2020]. O Scrum é baseado no processo Lean utilizado pela Toyota [Schwaber and Sutherland 2020, Sutherland 2014], que reduz o desperdício, se concentra no essencial, aumentando a velocidade da entrega e qualidade do produto, gerando mais valor para o cliente [Sutherland 2014].

O 15º relatório anual *State of Agile*, responsável por detalhar tendências e problemas notáveis na adoção de técnicas e prática ágeis, indica um crescimento significativo na adoção dos métodos ágeis de 37% em 2020 para 86% em 2021 nas equipes de desenvolvimento de software [Digital.ai 2021]. Essa mesma pesquisa destaca que o Scrum é a abordagem ágil mais popular com 66% de aprovação, e suas derivações com 15% [Digital.ai 2021]. A importância na escolha de uma ferramenta de gestão de projeto, pois a ferramenta fornece soluções para gerir, planejar e acompanhar o projeto de desenvolvimento de software [Flores 2022]. Porém, a adoção de uma metodologia ágil é dificultada pela necessidade de escolha entre diferentes ferramentas de software aplicáveis.

A seleção de ferramentas não é uma atividade fácil, pois existem várias opções que implementam as metodologias ágeis, mas não em sua completude. Muitas dessas ferramentas já estão consolidadas no mercado e apresentam funcionalidades que acomodam desde os projetos mais simples aos mais complexo, por isso a escolha da melhor opção se torna um ponto de extrema importância para a execução do projeto, reduzindo as chances de mudança da ferramenta no andamento do projeto e evitando possíveis contratempos para a entrega do produto.

Considerando as dificuldades encontradas pelos gestores na escolha de uma ferramenta de gerenciamento de projeto que suporte a aplicação dos métodos ágeis, este trabalho tem como objetivo geral indicar como as ferramentas de gerenciamento de projeto implementam metodologias ágeis. Para alcançá-lo foram definidos os seguintes objetivos específicos: escolher a metodologia ágil que será investigada; identificar ferramentas de

gerenciamento de projeto mais utilizadas; definir um conjunto de características e métricas das metodologias escolhidas; e analisar se as ferramentas selecionadas estão em conformidade com as características das metodologias ágeis investigadas.

Este artigo está organizado da seguinte forma: a Seção 2 traz a referencial teórico; a Seção 3 trata dos trabalhos relacionados; a Seção 4 trata do método de pesquisa utilizado; a Seção 5 traz os resultados; e a Seção 6 traz a conclusão; e a Seção 7 traz o apêndice.

2. Referencial Teórico

2.1. Gestão de Projetos

Projetos são empreendimentos com objetivo e prazo definidos, em detrimento da limitação de recursos [Cavalcanti 2016]. Todo projeto requer um esforço temporário empreendido para criar um produto, serviço ou resultado único [PMI 2017].

Nos projetos, como uma organização (articulação) de um conjunto de pessoas que visam um objetivo específico e em comum, com datas de início e fim bem definidas, são envolvidos: gastos, ações ou empreendimentos com alto risco, além de recursos necessários para sua execução. Diante disso, a conclusão do projeto só ocorre quando os objetivos e requisitos são alcançados em detrimento de todas as limitações impostas anteriormente mencionadas [Carvalho 2019]. Como alternativa para alcançar-se tal sucesso se faz necessário a gestão de projetos.

A gestão de projeto é a área da administração na qual o gerenciamento do projeto tem como foco o planejamento, organização, supervisão e controle sobre aspectos relacionados ao desenvolvimento do projeto [Carvalho 2019]. Esses aspectos são o escopo, o tempo, o custo, e a qualidade [Cavalcanti 2016], cuja gestão integram conhecimento, habilidade, técnicas e ferramentas a fim de atender os requisitos do projeto [PMI 2017], para satisfazer as necessidades das partes interessadas (*stakeholders*), ou até superando as mesmas. Desta forma, a gestão de projeto funcionará como uma facilitadora na identificação das necessidades, definição de objetivos de acordo com a prioridade de execução, satisfazendo as expectativas dos *stakeholders* [Cavalcanti 2016].

Na década de 1960, surgem as associações *Project Management Institute* (Instituto de Gerenciamento de Projetos ou PMI) nos Estados Unidos e a *International Project Management Association* (Associação Internacional de Gerenciamento de Projetos ou IPMA) em Viena na Áustria [Carvalho 2019]. Em 1970, a área de gestão de projetos foi vista como um mercado promissor [Carvalho 2019], impulsionado por software específicos de planejamento e controle, que mesmo com algumas restrições auxiliou no apoio da tomada de decisão [Cotas 1987]. Nas décadas de 1980 e 1990, houve a consolidação das boas práticas de gerenciamento de projeto. Na segunda metade da década de 1990, houve um crescimento acelerado da área de gerenciamento de projeto, muito motivado pelo crescimento de associações e institutos, foi nesse período que houve a publicação do primeiro guia de gerenciamento de projeto o *Project Management Body of Knowledge* (PMBOK) e a certificação dos profissionais da área *Project Management Professional* (PMP) [Carvalho 2019]. Esse alto crescimento consolidou o gerenciamento de projeto como metodologia, passando a ser disciplina obrigatória nas empresas que querem desenvolver e manter

vantagens competitivas, sendo mencionado por diversos estudiosos da administração [Carvalho 2019], tornando-se uma opção de formação complementar entre os profissionais das áreas de negócio que são baseadas em projetos nos setores de produtos, serviços e inovação [Cavalcanti 2016]. A partir do início dos anos 2000, o foco da área passou para os modelos organizacionais, como forma de gerir mudanças no processo organizacional e a estruturar os planos de ação em gestão de projetos [Carvalho 2019], iniciando a corrida em direção ao gerenciamento de projetos. Isso gerou a mudança do perfil do gerente de projeto, o qual era eminente técnico e se passou para uma abordagem mais gerencial [Carvalho 2019].

No início, o gerenciamento de projeto era adotado principalmente por administrações de grandes empreiteiras e órgãos governamentais, responsáveis por projetos de médio prazo, entre 3 e 5 anos [Carvalho 2019]. Com o tempo essa forma de gerenciamento passou a ser utilizada em outras áreas como na área de tecnologia da informação e comunicação (TICs), no processo de desenvolvimento de software [Cavalcanti 2016].

Nesse contexto de TIC, atualmente são utilizados com frequência os métodos ágeis de gestão de projetos.

2.2. Gestão Ágil de Projeto

O gerenciamento ágil de projetos é uma abordagem diferente da gestão de projetos tradicional definida no PMBoK, que tem como objetivo otimizar o processo de desenvolvimento e entrega de um projeto [Carvalho 2019]. Essa é uma abordagem que surgiu na área de Tecnologia da Informação para o processo de desenvolvimento de softwares, guiada pelo manifesto ágil [Beck et al. 2001]. O manifesto ágil surgiu em 2001, a partir de uma reunião de líderes de desenvolvimento de softwares [Sutherland 2014], para resolver os problemas no desenvolvimento de softwares que utilizam o gerenciamento de projetos tradicionais.

O processo de desenvolvimento demanda muito tempo, elevado custo de desenvolvimento e softwares criados não satisfazendo as necessidades dos *stakeholders* ou depois de ficarem prontos, já estarem obsoletos [Sutherland 2014]. Outra dificuldade enfrentada era a mudança de requisitos do software pelos *stakeholders*. Para acomodar essa mudança, em algumas situações era necessário executar novamente todas as etapas do processo de desenvolvimento de software, ou, em alguns casos, os novos requisitos eram até ignorados, por não fazer parte do escopo inicial do projeto. O manifesto ágil contém quatro valores [Beck et al. 2001], sendo eles:

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

Na abordagem ágil, as entregas de partes funcionais do software que está sendo desenvolvido acontecem com frequência. Essas partes são desenvolvidas em várias iterações (ciclos), implementando os requisitos mais importantes para o cliente. Isso permite a identificação mais rápida de inconsistências e problemas de qualidade no software que está sendo desenvolvido quanto no seu processo de desenvolvimento [Sommerville 2018].

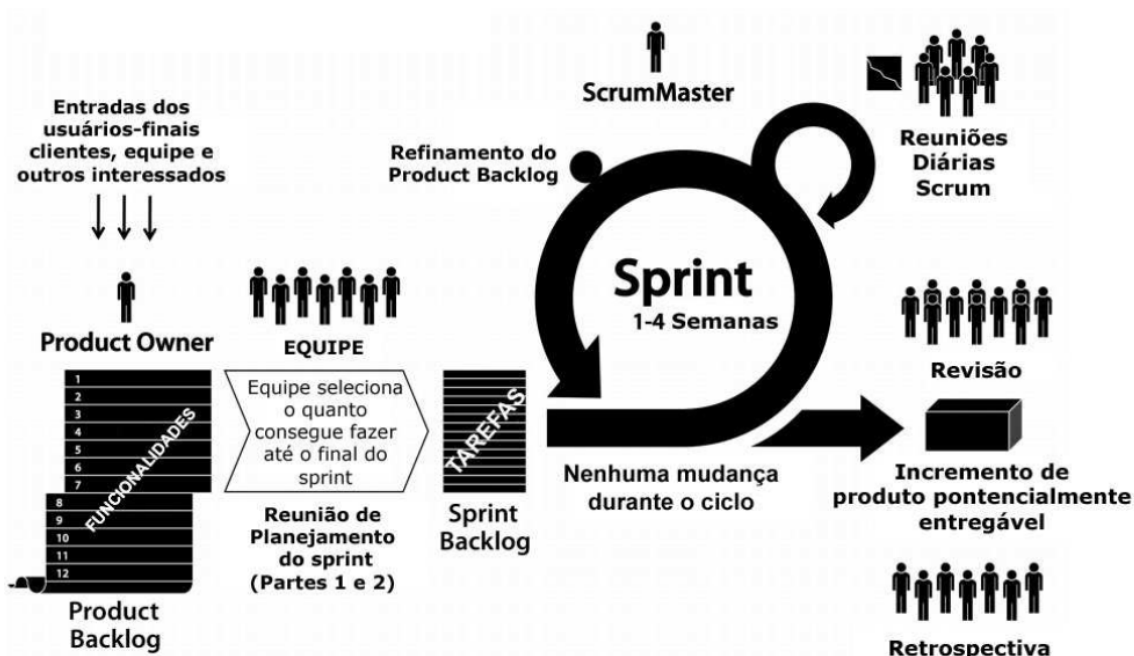
Nesse cenário, os objetivos do cliente são priorizados e os itens de maior valor para o negócio são implementados primeiro. A prioridade é definida por um representante do negócio, com a ajuda da equipe do projeto, promovendo o engajamento dos membros da equipe com as partes interessadas chave. Dessa forma, é maximizada a satisfação do cliente ao longo do projeto e a equipe responsável pela criação do software pode reagir melhor aos altos níveis de mudanças nos requisitos [PMI 2017, Sommerville 2018]. Há várias abordagens de gestão ágil de projetos, dentre elas o Scrum e o Kanban.

2.3. Scrum

A versão inicial do Scrum foi criada no início dos anos 90, por Ken Schwaber e Jeff Sutherland [Beck et al. 2001]. Esse acontecimento foi uma mudança radical em relação às metodologias prescritivas e hierarquizadas utilizadas no gerenciamento de projetos naquela época, pois o Scrum é uma metodologia de desenvolvimento mais rápida, confiável e eficiente [Sutherland 2014].

O Scrum define uma forma de trabalhar que gera valor por meio de soluções adaptativas para problemas complexos, assim ajudando pessoas, times de trabalho e organizações [Schwaber and Sutherland 2020]. Ele é baseado no processo Lean utilizado pela Toyota, que reduz o desperdício e se concentra no essencial para alcançar os objetivos estabelecidos para o software que será criado [Schwaber and Sutherland 2020, Sutherland 2014]. O resumo pode ser visto na Figura 1.

Figura 1. Processo Scrum



Fonte: [Sutherland 2014]

A estrutura do Scrum é dividida em três grupos, sendo eles: Equipes Scrum; Eventos Scrum; e Artefatos do Scrum.

A equipe Scrum: A equipe Scrum consiste em pequenos times, multifuncionais e auto-gerenciáveis. O time é responsável por todas as atividades para criar incrementos que adicionem valor ao software que está sendo criado a cada Sprint (ciclo de desenvolvimento que gera uma parte do software global). A equipe Scrum é constituída por três papéis:

- *Scrum Master* líder do time, responsável por estabelecer a metodologia Scrum, garantir sua correta execução e pela eficiência da equipe Scrum.
- *Product Owner* (PO) responsável por maximizar o valor do produto resultante do trabalho da equipe Scrum e gerenciar o *Backlog* do Produto (requisitos do cliente).
- Desenvolvedores, as pessoas da equipe Scrum que estão comprometidas em criar um incremento utilizável a cada Sprint, parte funcional do produto global que será criado [Schwaber and Sutherland 2020].

Eventos Scrum: Os eventos Scrum são oportunidades para inspecionar e adaptar os artefatos do Scrum. Eles são utilizados para criar regularidade e minimizar a necessidade de reuniões (não previstas na metodologia) [Schwaber and Sutherland 2020]. O principal evento Scrum é a Sprint. As Sprints são ciclos incrementais de duração fixa durante o desenvolvimento [Schwaber and Sutherland 2020]. Essa duração pode ser de um mês ou menos. Durante a Sprint um incremento é desenvolvido, ou seja, uma parte funcional do software que será criado é desenvolvido e, no final da Sprint, ele é validado pelo cliente [Schwaber and Sutherland 2020]. É importante mencionar que a Sprint agrupa outros quatro eventos do Scrum:

- Planejamento da Sprint evento no qual é definido o conjunto de requisitos que será desenvolvido na Sprint e as atividades necessárias para transformar esses requisitos em uma parte funcional do software que está sendo criado [Schwaber and Sutherland 2020]. Esse planejamento é feito com a colaboração de todos da Equipe Scrum [Schwaber and Sutherland 2020].
- Reunião Diária evento que tem como propósito inspecionar o progresso da Sprint em direção ao objetivo da Sprint e adaptar o Sprint *Backlog* conforme necessário, ajustando o próximo trabalho planejado para próxima Sprint [Schwaber and Sutherland 2020].
- Revisão da Sprint evento realizado ao final da Sprint, para inspecionar o resultado da Sprint (o incremento) e determinar as futuras adaptações necessárias nele [Schwaber and Sutherland 2020].
- Retrospectiva da Sprint evento que tem como propósito colher feedback e planejar maneiras de aumentar a qualidade e a eficácia da próxima Sprint.

Artefatos do Scrum: Os artefatos são informações usadas pela equipe Scrum e os *stakeholders* para detalhar o produto desenvolvido durante o projeto [Drumond 2021 , Schwaber and Sutherland 2020]. Existem os seguintes artefatos: *Backlog* do Produto, *Backlog* da Sprint e Incremento.

- *Backlog* do Produto: Lista de requisitos necessários para melhorar o software que está sendo criado. Essa lista é ordenada pelo *Product Owner* de acordo com o que é mais

importante para o cliente e deve ser implementado primeiro [Schwaber and Sutherland 2020].

- *Sprint Backlog*: Artefato composto pela Meta da Sprint, o conjunto de itens do *Backlog* do Produto selecionados para a Sprint e um plano de ação para entregar o incremento. Esse plano é feito por e para os Desenvolvedores [Schwaber and Sutherland 2020].
- *Incremento*: Resultado obtido das tarefas do *Backlog* do Produto [Drumond 2021]. Para alcançar a Meta do Produto, os incrementos são adicionados aos incrementos anteriores e completamente verificado, garantindo que todos os incrementos funcionem juntos, a fim de fornecer valor para o negócio [Schwaber and Sutherland 2020]. É importante notar que o incremento deve ser utilizável [Schwaber and Sutherland 2020].

2.4. Kanban

O Kanban é um método ágil com abordagem menos prescritiva, comparado aos outros métodos baseado nos princípios do Desenvolvimento Lean de Produtos [Boeg 2010].

Kanban é uma palavra japonesa que significa cartão visual. O termo é recente na área de TI, mas foi utilizado em produção Lean na Toyota por mais de 50 anos, para controlar e equilibrar a linha de produção visualmente [Boeg 2010]. O uso do Kanban é muito útil para visualizar e otimizar o fluxo de trabalho, e tem se tornado bastante popular como extensão de outros métodos ágeis [Boeg 2010].

No Kanban, se destacam cinco princípios: Visualizar o Fluxo de Trabalho, Limitar o Trabalho em Progresso (WIP – Work in Progress), Medir e Gerenciar o Fluxo, Tornar as Políticas do Processo Explícitas e Usar Modelos para Reconhecer Oportunidades de Melhoria [Anderson et al. 2011].

- *Visualizar o Fluxo de Trabalho*: Uma boa visualização é a chave para colaborar efetivamente e identificar oportunidades de melhoria, pois melhora muito mais a clareza do trabalho e do seu fluxo [University 2021]. Isso permite que você absorva e processe muitas informações em um curto espaço de tempo. O quadro Kanban é o meio mais comum de visualizar um sistema Kanban, mover o trabalho da esquerda para a direita é comum em todos os quadros [University 2021]. Os novos itens de trabalho entram no quadro pela esquerda, quando os itens agregam valor para o cliente, eles saem pela direita [University 2021].

Em um sistema Kanban, há uma representação da quantidade de trabalho permitida (trabalho em progresso, WIP). Os itens de trabalho podem ser de vários tipos e tamanhos, de tarefas a requisitos [University 2021]. Os itens de trabalho geralmente aparecem em notas individuais, geralmente chamadas de cartões ou sticky notes [University 2021]. A sequência de atividades pelas quais esses itens de trabalho passam é chamada de fluxo de trabalho. As etapas no fluxo de trabalho são exibidas em colunas [University 2021]. A Figura 2 ilustra um quadro Kanban simplificado.

- *Limites de Trabalho em Progresso (WIP)*: O WIP indica o número de itens de trabalho em progresso em um determinado momento. Usando Kanban, descobrimos que

sistemas eficazes são mais focados no fluxo de trabalho e menos no uso do desenvolvedor [University 2021]. Quando os recursos são totalmente utilizados, o sistema não cede e o resultado é um rendimento muito baixo. No Kanban, limitamos o WIP para equilibrar o uso e garantir o fluxo de trabalho.

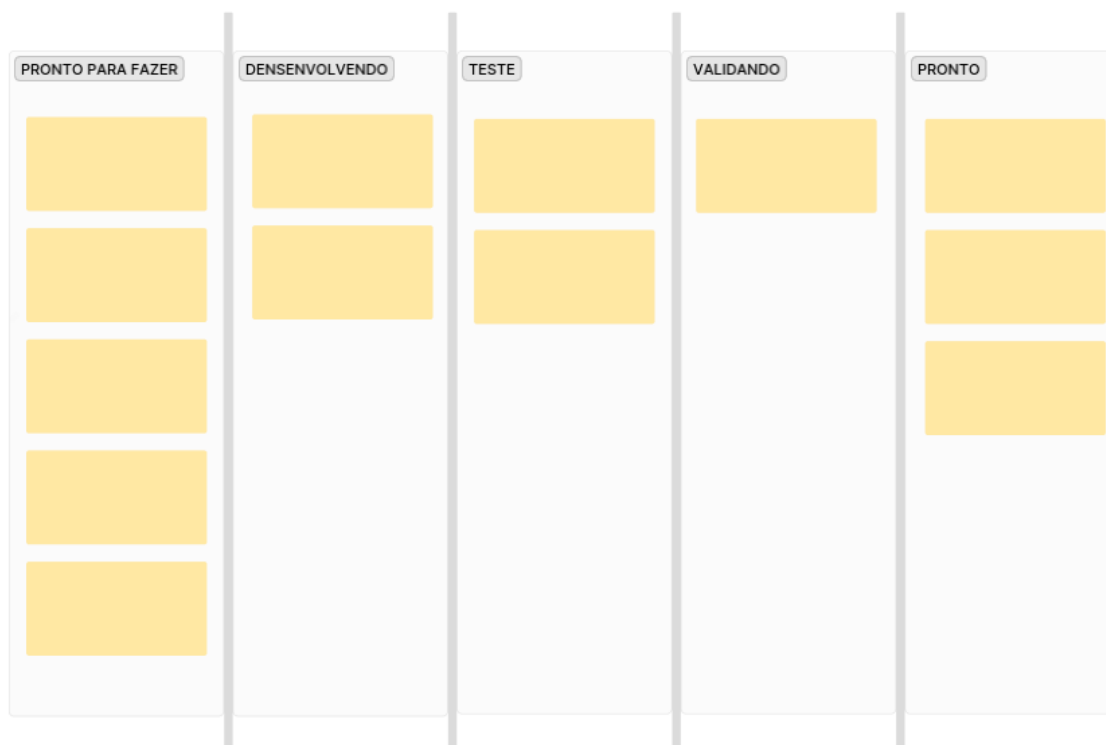
- Medir e Gerenciar o Fluxo: O objetivo do gerenciamento de fluxo de trabalho é fazer as coisas da maneira mais perfeita e previsível possível, mantendo um ritmo constante [University 2021]. Como mencionado anteriormente, a regulação do WIP é uma das principais maneiras de garantir um fluxo suave e previsível. O acompanhamento ou medição do fluxo de trabalho gera informações críticas que são úteis no gerenciamento das expectativas, previsões e melhorias do cliente [University 2021].
- Tornar as Políticas Explícitas: As políticas explícitas, permite que novos colaboradores entendam rapidamente como o trabalho deve ser organizado [University 2021]. Estas políticas incluem:
 - Políticas como o de reabastecimento do quadro (quando, quanto, por quem);
 - Definição de quando uma atividade de trabalho é concluída, e o item de trabalho pode seguir em frente (“critérios para puxar”);
 - Limites de WIP;
 - Políticas para o tratamento de itens de trabalho de diferentes classes de serviço;
 - Horários das reuniões e conteúdo;
 - Outros princípios e acordos de colaboração.

As políticas devem permitir a auto-organização dentro do grupo de pessoas que executam o método Kanban [University 2021]. As políticas devem ser: Escassas (poucas), Simples, Bem definidas, Visíveis, Aplicáveis e Facilmente alteradas pelos prestadores de serviços [University 2021].

- Usar Modelos para Reconhecer Oportunidades de Melhoria: Voltando aos princípios da gestão de mudanças, no método Kanban “Comece com o que você faz hoje” e “Concorde em buscar a melhoria através da mudança evolutiva”. Kanban é um método de mudança contínua, e fazemos essas mudanças de forma colaborativa usando experimentos orientados a modelos e o método científico [University 2021]. O *feedback* e as métricas de *lead time* (tempo de provisionamento) e *throughput* (vazão) são cruciais para nos guiar [University 2021]. São criados experimentos seguros para que, se a hipótese estiver correta e o experimento der bons resultados, a mudança seja mantida, mas se não, é possível voltar ao estado anterior facilmente [University 2021].

Os princípios mencionados levam os especialistas a concordarem que o Kanban é um método de mudança e que estimula a cultura da melhoria contínua nas organizações [Boeg 2010].

Figura 2. Quadro Kanban



Fonte: Elaborado pelos autores

3. Trabalhos Relacionados

[Feltrin 2012] fez uma estudo para realizar uma avaliação comparativa entre ferramentas Open Source para o suporte ao desenvolvimento ágil baseados nos princípios do método ágil Scrum. Foram selecionadas as ferramentas: Agilefant, Kunagi e Firescrum. O autor utilizou a norma NBR ISO/IEC 14598-1, para obter requisitos e recomendações para implementação prática da avaliação de um produto de software e a norma NBR ISO/IEC 9126-1 para selecionar características de qualidade para comparar as ferramentas em relação ao Scrum e identificar as que adotam mais princípios do Scrum (atendem mais aos requisitos de qualidade ligados ao princípios do Scrum). Como resultado, a ferramenta Kunagi obteve o maior índice de atendimento dos requisitos de qualidade, aproximadamente 86%, ainda segundo o autor, é possível afirmar que, as ferramentas *Open Source* avaliadas estão acima dos 70% dos requisitos avaliados.

[Soares 2017] realizou uma análise comparativa entres as ferramentas que adotam o método ágil Kanban, como ferramentas para controle de cadência de processo e aumento de vazão, no âmbito de desenvolvimento de software. Para isso o autor fez um estudo sobre as propriedades básicas do Kanban como método de desenvolvimento de software e realizou a seleção das ferramentas de acordo com seus trabalhos relacionados: Trello, Jira, CA Agile Central e LeanKit. Os critérios de comparação das ferramentas foram definidos de acordo com as propriedades do Kanban e requisitos relacionados com o uso do Kanban como ferramenta de auxílio para o desenvolvimento de software. Para a avaliação, foi reproduzido um quadro Kanban, a investigação do uso das métricas e a implementação de suas funcionalidades, e foi feita a análise comparativa entre as ferramentas de acordo com os

critérios implementados. Como resultado, as ferramentas CA Agile Central e LeanKit cumprem todos os requisitos, enquanto Trello e Jira cumprem apenas os requisitos de desenvolvimento de software.

[Carvalho et al. 2020] investigaram aplicativos móveis de gerenciamento de projetos no âmbito nacional, iniciando pela identificação dos aplicativos mais utilizados, e de quais grupos de processo e área do conhecimento do PMBOK os mesmos são aplicados. Os autores realizaram uma pesquisa de opinião com gerentes de projetos e foi realizada uma análise quantitativa e qualitativa. A análise quantitativa mostrou quais os aplicativos mais utilizados e que a maioria dos gerentes de projetos não utilizam. Na análise qualitativa, utilizando os procedimentos de Teoria Fundamentada em Dados, foi possível identificar motivos relacionados à ferramenta, à empresa, ao projeto e à equipe pelo não uso das ferramentas, alguns dos motivos identificados na análise são: falta de interesse no uso da ferramenta, falta de conhecimento, limitações dos aplicativos e complexidade dos projetos.

Tabela 1. Comparativo entre os trabalhos relacionados

Crítérios	[Carvalho et al. 2020]	[Feltrin 2012]	[Soares 2017]	Trabalho Atual
Unidade de análise	Método tradicional PMBoK	Método ágil Scrum	Método ágil Kanban	Métodos ágeis mais utilizados atualmente
Tipo de software avaliado	Aplicativos móveis	Sistemas desktop e web ambos open source	Sistemas web	Aplicativo móveis
Norma usada para avaliação da qualidade	Não se aplica	ISO/IEC 14598-1 ISO/IEC 9126-1	Não se aplica	ISO/IEC 25040 ISO/IEC 25010
Características avaliadas (Crítérios de comparação)	Não se aplica	Funcionalidade Confiabilidade Usabilidade Eficiência Manutenibilidade Portabilidade	Não se aplica	Funcionalidade

Fonte: Elaborado pelos autores

4. Método

4.1. Seleção de Métodos Ágeis e Ferramentas de Software

Foram escolhidos os métodos ágeis Scrum e Kanban, pois são as abordagens ágeis mais populares e utilizadas atualmente na indústria [Digital.ai 2021]. Foram selecionadas ferramentas do tipo *mobile applications* (*App*). Elas são softwares criados para utilização em celulares e tablets. Os aplicativos selecionados foram o Trello e a Asana, que são ferramentas de gestão de projetos e estão entre as ferramentas mais baixadas desta categoria no Google Play Store. O Trello possui mais de 10 milhões de downloads e a Asana possui mais de 5 milhões. Esses aplicativos permitem o acompanhamento do fluxo de trabalho para uma compreensão clara das atividades. Foram escolhidos aplicativos para analisar, porque nos últimos anos foi intensificada a execução remota de atividades de trabalho e o *App* oferece maior flexibilidade de uso remoto [Manole and Avramescu 2017]. Além disso, é importante

avancar no estudo da adesão dessas ferramentas aos métodos ágeis selecionados para análise, uma vez que eles são amplamente usados atualmente.

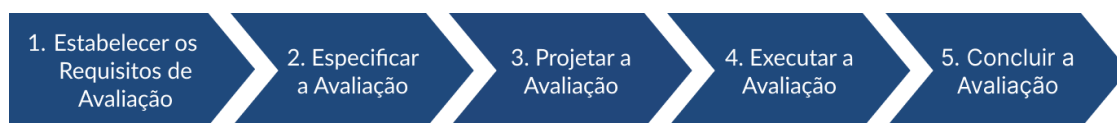
Nesse contexto, foi feita uma análise com base em seu nível de utilização no mercado e disponibilidade para baixar e usar as duas ferramentas. Depois de selecionar os métodos ágeis que foram analisados e os *Apps* de gestão ágil de projetos de software, foi executada a segunda etapa desta pesquisa (avaliação da qualidade). Nessa etapa, foi verificado o quanto os *Apps* investigados implementam os dois métodos ágeis escolhidos.

4.2. Avaliação da Qualidade

Os *Apps* foram avaliados de acordo com as etapas definidas pela ISO/IEC 25040 [ISO/IEC 25040 2011], com as devidas adaptações de acordo com as metodologias investigadas, processo similar ao utilizado por [Feltrin 2012]. Além disso, foi utilizado como base da avaliação das ferramentas o modelo de qualidade do produto de software proposto na ISO/IEC 25010, mais especificamente a característica da qualidade chamada "Adequação da Funcionalidade", pois ela possibilita avaliar o quanto as funcionalidades de um software estão adequadas às necessidades do cliente [ISO/IEC 25010 2011].

O processo de avaliação de qualidade definido pela ISO/IEC 25040, define cinco etapas principais [ISO/IEC 25040 2011], ilustradas a seguir:

Figura 3. Etapas do Processo de Avaliação



Fonte: Elaborado pelos autores

4.2.1. Estabelecer os Requisitos de Avaliação

A avaliação dos *Apps* em relação à adequação ao Scrum e Kanban foi realizada com base nas características e subcaracterísticas da norma ISO/IEC 25010, exibidas na Tabela 2, pois servem para avaliar a qualidade de produtos de software. A medição dessas características e suas respectivas subcaracterísticas para os aplicativos analisados foi realizada através dos requisitos funcionais indicados na Tabela 3. Esses requisitos foram definidos de acordo com os trabalhos de [Feltrin 2012] e [Soares 2017] e servem como critérios para avaliar os *Apps*. Após estabelecer os requisitos para avaliar os *Apps*, foi executada a avaliação das ferramentas de acordo com os critérios (requisitos funcionais) escolhidos.

Tabela 2. Características Adequação Funcional e suas subcaracterísticas no modelo de Qualidade de Produto de Software da ISO/IEC 25010

Características	Subcaracterísticas
Adequação Funcional	Completeza Funcional
	Correção Funcional
	Funcionalidade Apropriada

Fonte: Elaborado pelos autores

Tabela 3. Critérios utilizados para a avaliação dos aplicativos em relação aos métodos Scrum e Kanban

Método	Requisitos Funcionais (critérios de avaliação)
Scrum	Planning Poker
	Criar e manter Product Backlog
	Atribuir prioridades para os itens do Product Backlog
	Estimar tempo de execução para os itens do Product Backlog
	Atribuir responsáveis para os itens do Product Backlog
	Criar e manter um Sprint
	Criar e manter uma lista de impedimentos
	Emissão de relatório de atividades do Product Backlog
	Emissão de relatório de atividades do Sprint Backlog
	Emissão de relatório/gráfico Sprint Burndown
	Velocidade da Equipe
	Gráfico de Burndown
Kanban	Decomposição do processo em diversas etapas, mostradas no quadro de Kanban
	Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes
	Limitação do WIP para cada coluna do Kanban
	Lead Time Distribution
	Gráfico de Fluxo Cumulativo (GFC)
	Medição do Lead Time
	Medição do Throughput
Scrum e Kanban	Cycle Time
	Taxa de Entrega
Nenhum método	Exportar o projeto
	Importar o projeto
	Meio de comunicação entre os usuários
	Criar e manter mais de um projeto
	Permite customização do tipo de acesso do usuário
	Interage com outros sistemas
	Criação de diferentes projetos e times
	Separação dos itens entre os projetos ou times
	Capacidade de descrever os itens de trabalho
	Atribuição de responsável para cada item
	Interação entre membros do time de desenvolvimento em relação a um item de trabalho
	Priorização dos itens de trabalho

Fonte: Elaborado pelos autores

4.2.2. Especificar a Avaliação

Nesta etapa, os requisitos funcionais (critérios de comparação dos aplicativos) definidos na etapa anterior foram classificados, pelos pesquisadores e com base em [Feltrin 2012] e [Soares 2017], nas seguintes subcaracterísticas da ISO/IEC 25010: completeza funcional, correção funcional e funcionalidade apropriada, que fazem parte da característica principal chamada “Adequação Funcional”.

Tabela 4. Critério de avaliação classificados de acordo com a subcaracterística Completeza Funcional

Características	Subcaracterísticas	Descrição
Adequação Funcional	Completeza Funcional	Planning Poker
		Atribuir prioridades para os itens do Product Backlog
		Estimar tempo de execução para os itens do Product Backlog
		Atribuir responsáveis para os itens do Product Backlog
		Decomposição do processo em diversas etapas, mostradas no quadro de Kanban
		Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes
		Exportar o projeto
		Importar o projeto
		Criar e manter mais de um projeto

Fonte: Elaborado pelos autores

Tabela 5. Critério de avaliação classificados de acordo com a subcaracterística Correção Funcional

Características	Subcaracterísticas	Descrição
Adequação Funcional	Correção Funcional	Criar e manter Product Backlog
		Criar e manter um Sprint
		Criar e manter uma lista de impedimentos
		Limitação do WIP para cada coluna do Kanban
		Permite customização do tipo de acesso do usuário

Fonte: Elaborado pelos autores

Tabela 6. Critério de avaliação classificados de acordo com a subcaracterística Funcionalidade Apropriada

Características	Subcaracterísticas	Descrição
Adequação Funcional	Funcionalidade apropriada	Emissão de relatório de atividades do Product Backlog
		Emissão de relatório de atividades do Sprint Backlog
		Emissão de relatório/gráfico Sprint Burndown
		Velocidade da Equipe
		Gráfico de Burndown
		Lead Time Distribution
		Lead Time Distribution
		Gráfico de Fluxo Cumulativo (GFC)
		Medição do Lead Time
		Medição do Throughput
		Cycle Time
		Taxa de Entrega
		Meio de comunicação entre os usuários
		Interage com outros sistemas
		Criação de diferentes projetos e times
		Separação dos itens entre os projetos ou times
		Capacidade de descrever os itens de trabalho
		Atribuição de responsável para cada item
		Interação entre membros do time de desenvolvimento em relação a um item de trabalho
		Priorização dos itens de trabalho

Fonte: Elaborado pelos autores

Em seguida, foi definida a escala de avaliação dos critérios e seus possíveis pesos, como indicado nas Tabelas 7 e 8, respectivamente. É importante destacar que o peso de cada critério pode ser atribuído pelo avaliador, segundo sua percepção de importância do critério. Porém, os pesos e a escala utilizadas aqui foram obtidas

Tabela 7. Escala do nível de atendimento das métricas

Escala da métrica	
Descrição	Valor
Não atende	0
Atende Parcialmente	1
Atende	2

Fonte: Elaborado pelos autores

Tabela 8. Peso do grau de importância das métricas

Peso da métrica	
Descrição	Valor
Essencial	3
Importante	2
Desejável	1

Fonte: Elaborado pelos autores

Tabela 9. Atribuição do grau de importância para os critérios avaliativos relacionados às subcaracterísticas completude funcional

Características	Subcaracterísticas	ID	Descrição	Pesos
Adequação funcional	1. Completude funcional	1.1	Planning Poker	1
		1.2	Atribuir prioridades para os itens do Product Backlog	2
		1.3	Estimar tempo de execução para os itens do Product Backlog	2
		1.4	Atribuir responsáveis para os itens do Product Backlog	2
		1.5	Decomposição do processo em diversas etapas, mostradas no quadro de Kanban	3
		1.6	Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes	2
		1.7	Exportar o projeto	1
		1.8	Importar o projeto	1
		1.9	Criar e manter mais de um projeto	1

Fonte: Elaborado pelos autores

Tabela 10. Atribuição do grau de importância para os critérios avaliativos relacionados às subcaracterísticas correção funcional

Características	Subcaracterísticas	ID	Descrição	Pesos
Adequação funcional	2. Correção funcional	2.1	Criar e manter Product Backlog	3
		2.2	Criar e manter um Sprint	3
		2.3	Criar e manter uma lista de impedimentos	2
		2.4	Limitação do WIP para cada coluna do Kanban	3
		2.5	Permite customização do tipo de acesso do usuário	1

Fonte: Elaborado pelos autores

Tabela 11. Atribuição do grau de importância para os critérios avaliativos relacionados às subcaracterísticas funcionalidade apropriada

Características	Subcaracterísticas	ID	Descrição	Pesos
Adequação funcional	3. Funcionalidade apropriada	3.1	Emissão de relatório de atividades do Product Backlog	1
		3.2	Emissão de relatório de atividades do Sprint Backlog	1
		3.3	Emissão de relatório/gráfico Sprint Burndown	1
		3.4	Velocidade da Equipe	3
		3.5	Gráfico de Burndown	2
		3.6	Lead Time Distribution	3
		3.7	Gráfico de Fluxo Cumulativo (GFC)	3
		3.8	Medição do Lead Time	1
		3.9	Medição do Throughput	1
		3.10	Cycle Time	2
		3.11	Taxa de Entrega	1
		3.12	Meio de comunicação entre os usuários	2
		3.1	Interage com outros sistemas	2
		3.2	Criação de diferentes projetos e times	1
		3.3	Separação dos itens entre os projetos ou times	1
		3.4	Capacidade de descrever os itens de trabalho	2
		3.5	Atribuição de responsável para cada item	2
		3.6	Interação entre membros do time de desenvolvimento em relação a um item de trabalho	2
		3.7	Priorização dos itens de trabalho	2

Fonte: Elaborado pelos autores

Por fim, foi definido como obter o desempenho de cada App em cada critério. Esse desempenho foi determinado pela fórmula $X Pi = V \times P$, onde Pi indica a pontuação total do item, V o valor atribuído à métrica e P o peso da métrica. Após a obtenção da pontuação total do item, foi realizado o cálculo da pontuação final de cada App utilizando a seguinte

fórmula $Pf = \sum_{i=1}^n Pi$, onde Pf é a pontuação total final; n a quantidade de critérios; i o número do critério, variando de 1 a n ; e Pi a pontuação total do item. Ao fim, seu valor numérico serviu para comparar os Apps entre si.

4.2.3. Projetar a Avaliação

Nesta etapa foi produzido um plano de avaliação, ou seja, foi criado um *script* [Feltrin 2012] baseado no trabalho de e adaptado ao presente trabalho, este *script* contém o passo a passo de como executar a avaliação dos *Apps*, indicando os procedimentos que foram executados para efetuar a análise de cada software.

4.2.4. Executar a Avaliação

Nesta etapa foi realizada a avaliação dos *Apps* de acordo com os seguintes critérios da ISO/IEC 25010 relacionados aos requisitos funcionais do produto de software: completeza funcional, correção funcional e funcionalidade apropriada.

4.2.5. Concluir a Avaliação

Nesta etapa, foram analisados os resultados da avaliação e foi elaborado um relatório dela, apresentado na próxima seção (seção 5). Esse relatório contém a descrição do desempenho dos (*Apps*) em cada requisito funcional, segundo cada critério, além de observações sobre a avaliação, por exemplo: “*Funcionalidade não existente no aplicativo (FNE)*”. Essa observação indica que o *App* não possui uma funcionalidade necessária à execução do método ágil analisado.

5. Resultados

Após a conclusão de todas as etapas da avaliação, foi viável obter os resultados do nível de aderência de cada aplicativo aos métodos Scrum e Kanban. Isso pode ser verificado na Tabela 12 que corresponde ao *App* Trello e na Tabela 13 para a Asana. É importante destacar que as pontuações 1, 2 e 3 significam, respectivamente: 1) O aplicativo não oferece a funcionalidade de forma alguma; 2) A funcionalidade está disponível de forma parcial, requerendo adaptações ou extensões adicionais para atingir o resultado desejado; 3) A funcionalidade é nativa no aplicativo ou requer apenas alguns passos adicionais para ser habilitada.

Tabela 12. Relatório de avaliação para o *App* Trello

Subcaracterísticas	ID	Descrição	Pesos	Nível de atendimento	Obs.	Pontuação
Completeza Funcional	1.1	Planning Poker	1	0	FNE	0
	1.2	Atribuir prioridades para os itens do Product Backlog	2	1	-	2
	1.3	Estimar tempo de execução para os itens do Product Backlog	2	2	-	4
	1.4	Atribuir responsáveis para os itens do Product Backlog	2	2	-	4
	1.5	Decomposição do processo em diversas etapas, mostradas no quadro de Kanban	3	2	-	6
	1.6	Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes	2	0	FNE	0
	1.7	Exportar o projeto	1	0	FNE	0

	1.8	Importar o projeto	1	0	FNE	0
	1.9	Criar e manter mais de um projeto	1	2	-	2
Correção Funcional	2.1	Criar e manter Product Backlog	3	2	-	6
	2.2	Criar e manter um Sprint	3	1	-	3
	2.3	Criar e manter uma lista de impedimentos	2	1	-	2
	2.4	Limitação do WIP para cada coluna do Kanban	3	2	-	6
	2.5	Permite customização do tipo de acesso do usuário	1	2	-	2
Funcionalidade apropriada	3.1	Emissão de relatório de atividades do Product Backlog	1	0	FNE	0
	3.2	Emissão de relatório de atividades do Sprint Backlog	1	0	FNE	0
	3.3	Emissão de relatório/gráfico Sprint Burndown	1	0	FNE	0
	3.4	Velocidade da Equipe	3	0	FNE	0
	3.5	Gráfico de Burndown	2	0	FNE	0
	3.6	Lead Time Distribution	3	0	FNE	0
	3.7	Gráfico de Fluxo Cumulativo (GFC)	3	0	FNE	0
	3.8	Medição do Lead Time	1	0	FNE	0
	3.9	Medição do Throughput	1	0	FNE	0
	3.10	Cycle Time	2	0	FNE	0
	3.11	Taxa de Entrega	1	0	FNE	0
	3.12	Meio de comunicação entre os usuários	2	2	-	4
	3.13	Interage com outros sistemas	2	0	FNE	0
	3.14	Criação de diferentes projetos e times	1	1	-	1
	3.15	Separação dos itens entre os projetos ou times	1	1	-	1
	3.16	Capacidade de descrever os itens de trabalho	2	2	-	4
	3.17	Atribuição de responsável para cada item	2	2	-	4
	3.18	Interação entre membros do time de desenvolvimento em relação a um item de trabalho	2	2	-	4

	3.19	Priorização dos itens de trabalho	2	2	-	4
			Pontuação final			59

Fonte: Elaborado pelos autores

Tabela 13. Relatório de avaliação para o App Asana

Subcaracterísticas	ID	Descrição	Pesos	Nível de atendimento	Obs.	Pontuação
Completeza Funcional	1.1	Planning Poker	1	0	FNE	0
	1.2	Atribuir prioridades para os itens do Product Backlog	2	2	-	4
	1.3	Estimar tempo de execução para os itens do Product Backlog	2	2	-	4
	1.4	Atribuir responsáveis para os itens do Product Backlog	2	2	-	4
	1.5	Decomposição do processo em diversas etapas, mostradas no quadro de Kanban	3	2	-	6
	1.6	Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes	2	0	FNE	0
	1.7	Exportar o projeto	1	0	FNE	0
	1.8	Importar o projeto	1	0	FNE	0
	1.9	Criar e manter mais de um projeto	1	2	-	2
Correção Funcional	2.1	Criar e manter Product Backlog	3	1	-	3
	2.2	Criar e manter um Sprint	3	1	-	3
	2.3	Criar e manter uma lista de impedimentos	2	1	-	2
	2.4	Limitação do WIP para cada coluna do Kanban	3	0	FNE	0
	2.5	Permite customização do tipo de acesso do usuário	1	2	-	2
Funcionalidade apropriada	3.1	Emissão de relatório de atividades do Product Backlog	1	0	FNE	0
	3.2	Emissão de relatório de atividades do Sprint Backlog	1	0	FNE	0
	3.3	Emissão de relatório/gráfico Sprint Burndown	1	0	FNE	0
	3.4	Velocidade da Equipe	3	0	FNE	0
	3.5	Gráfico de Burndown	2	0	FNE	0

	3.6	Lead Time Distribution	3	0	FNE	0
	3.7	Gráfico de Fluxo Cumulativo (GFC)	3	0	FNE	0
	3.8	Medição do Lead Time	1	0	FNE	0
	3.9	Medição do Throughput	1	0	FNE	0
	3.10	Cycle Time	2	0	FNE	0
	3.11	Taxa de Entrega	1	0	FNE	0
	3.12	Meio de comunicação entre os usuários	2	2	-	4
	3.1	Interage com outros sistemas	2	0	FNE	0
	3.2	Criação de diferentes projetos e times	1	2	-	2
	3.3	Separação dos itens entre os projetos ou times	1	2	-	2
	3.4	Capacidade de descrever os itens de trabalho	2	2	-	4
	3.5	Atribuição de responsável para cada item	2	2	-	4
	3.6	Interação entre membros do time de desenvolvimento em relação a um item de trabalho	2	2	-	4
	3.7	Priorização dos itens de trabalho	2	2	-	4
			Pontuação final			54

Fonte: Elaborado pelos autores

A partir das avaliações realizadas nos dois *Apps*, foi possível realizar uma análise comparativa, o que permitiu chegar a uma conclusão sobre qual ferramenta proporciona o suporte mais eficaz na adoção dos métodos ágeis Scrum e Kanban, com base na característica Adequação Funcional da ISO/IEC 25010, que pode ser visto na Tabela 14.

Tabela 14. Análise comparativa entre os aplicativos avaliados com as pontuações finais obtidas para a característica Adequação Funcional

Características	Subcaracterísticas	Trello	Asana
Adequação funcional	Completeza Funcional	18	20
	Correção Funcional	19	10
	Funcionalidade apropriada	22	24
Pontuação Total		59	54

Fonte: Elaborado pelos autores

É importante mencionar que ao comparar os *Apps* usando apenas os critérios específicos da metodologia Scrum o aplicativo Trello obteve a melhor pontuação em relação ao Asana. Nesse caso, o Trello alcançou 21 pontos, enquanto o Asana obteve 20, conforme

exibido na Tabela 15. Isso indica que, em relação aos critérios específicos do Scrum, o Trello demonstrou maior compatibilidade e aderência em apoiar essa abordagem ágil de gerenciamento de projetos, mesmo que tenha demonstrado um desempenho ligeiramente superior comparado ao Asana.

Tabela 15. Análise comparativa dos aplicativos avaliados para a metodologia ágil Scrum

			Trello		Asana	
ID	Descrição	Pesos	Nível de atendimento	Pontuação	Nível de atendimento	Pontuação
1.1	Planning Poker	1	0	0	0	0
1.2	Atribuir prioridades para os itens do Product Backlog	2	1	2	2	4
1.3	Estimar tempo de execução para os itens do Product Backlog	2	2	4	2	4
1.4	Atribuir responsáveis para os itens do Product Backlog	2	2	4	2	4
2.1	Criar e manter Product Backlog	3	2	6	1	3
2.2	Criar e manter um Sprint	3	1	3	1	3
2.3	Criar e manter uma lista de impedimentos	2	1	2	1	2
3.1	Emissão de relatório de atividades do Product Backlog	1	0	0	0	0
3.2	Emissão de relatório de atividades do Sprint Backlog	1	0	0	0	0
3.3	Emissão de relatório/gráfico Sprint Burndown	1	0	0	0	0
3.4	Velocidade da Equipe	3	0	0	0	0
3.5	Gráfico de Burndown	2	0	0	0	0
Pontuação final			21		20	

Fonte: Elaborado pelos autores

Comparando os aplicativos usando apenas os critérios específicos da metodologia Kanban, o Trello obteve a melhor pontuação, alcançando 12 pontos, enquanto o Asana alcançou 6 pontos. Isso sugere que o Trello é mais adequado para a metodologia Kanban, como mostrado na Tabela a 16.

Tabela 16. Análise comparativa dos aplicativos avaliados para a metodologia ágil Kanban

			Trello		Asana	
ID	Descrição	Pesos	Nível de atendimento	Pontuação	Nível de atendimento	Pontuação
1.5	Decomposição do processo em diversas etapas, mostradas no quadro de Kanban	3	0	6	2	6
1.6	Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes	2	2	0	0	0
2.4	Limitação do WIP para cada coluna do Kanban	3	2	6	0	0
3.6	Lead Time Distribution	3	2	0	0	0
3.7	Gráfico de Fluxo Cumulativo (GFC)	3	2	0	0	0
3.8	Medição do Lead Time	1	0	0	0	0
3.9	Medição do Throughput	1	0	0	0	0
Pontuação final			12		6	

Fonte: Elaborado pelos autores

É relevante destacar que os critérios “Cycle Time” e “Taxa de Entrega” podem ser considerados como pertencentes às metodologias Scrum e Kanban. Nesse caso, elas não obtiveram pontuação nesses dois critérios, como pode ser observado na Tabela 17.

Tabela 17. Análise comparativa dos aplicativos avaliados para a metodologia ágil Kanban

			Trello		Asana	
ID	Descrição	Pesos	Nível de atendimento	Pontuação	Nível de atendimento	Pontuação
3.10	Cycle Time	2	0	6	0	0
3.11	Taxa de Entrega	1	0	0	0	0
Pontuação final			0		0	

Fonte: Elaborada pelo autor

Ao avaliar critérios gerais relacionados a gestão de projetos, desconsiderando os específicos da Scrum ou Kanban, o Trello obteve 26 pontos, enquanto o Asana alcançou 28 pontos. Isso destaca que o Asana demonstrou um desempenho ligeiramente superior considerando os critérios gerais de gestão de projetos, conforme apresentado na Tabela 18.

Tabela 18. Análise comparativa dos aplicativos avaliados para os critérios gerais de gestão de projetos

			Trello		Asana	
ID	Descrição	Pesos	Nível de atendimento	Pontuação	Nível de atendimento	Pontuação
1.7	Exportar o projeto	1	0	0	0	0
1.8	Importar o projeto	1	0	0	0	0
1.9	Criar e manter mais de um projeto	1	2	2	2	2
2.5	Permite customização do tipo de acesso do usuário	1	2	2	2	2
3.12	Meio de comunicação entre os usuários	2	2	4	2	4
3.13	Interage com outros sistemas	2	0	0	0	0
3.14	Criação de diferentes projetos e times	1	1	1	2	2
3.15	Separação dos itens entre os projetos ou times	1	1	1	2	2
3.16	Capacidade de descrever os itens de trabalho	2	2	4	2	4
3.17	Atribuição de responsável para cada item	2	2	4	2	4
3.18	Interação entre membros do time de desenvolvimento em relação a um item de trabalho	2	2	4	2	4
3.19	Priorização dos itens de trabalho	2	2	4	2	4
Pontuação final			28		26	

Fonte: Elaborada pelo autor

6. Conclusão

Após obtenção da pontuação total dos critérios avaliados e da realização da análise comparativa entre os aplicativos Trello e Asana, podemos concluir que o aplicativo Trello obteve a maior pontuação ao final da avaliação. Esta avaliação foi conduzida de acordo com a norma ISO/IEC 25040 e considerando os critérios de qualidade da característica "Adequação Funcional" da norma ISO/IEC 25010.

Neste contexto, este trabalho contribuiu para o mapeamento de ferramentas relevantes para o gerenciamento de projetos que utilizam alguma metodologia ágil analisadas neste estudo. Essa contribuição se dá pelo fato de ter sido realizada uma análise dos aplicativos de gestão ágil de projetos Trello e Asana, disponibilizando informações sobre a adesão (implementação) desses *Apps* aos métodos ágeis Scrum e Kanban. Dessa maneira é fornecida uma base de informações a serem utilizadas por empresas de desenvolvimento de software e gerentes de projetos, que utilizam o Scrum ou o Kanban, para identificar o aplicativo de gerenciamento de projetos mais adequado às suas necessidades.

Também é possível notar que, apesar do uso constante de smartphones e da migração de sistemas para suas plataformas, há uma falta de estudos com foco na utilização de aplicativos móveis para gestão de projetos e a preferência pela versão *web* dos aplicativos de gerenciamento de projeto, assim como foi apontado no trabalho de [Carvalho 2019]. Observou-se também que muitas das funcionalidades abordadas nos estudos de [Feltrin 2012]

e [Soares 2017] não estão presentes nos aplicativos Trello e Asana, ver Apêndice 1. Isso gera implicações tanto no uso desses aplicativos quanto na escolha da ferramenta mais apropriada para o gerenciamento de projeto.

A partir deste trabalho, alguns trabalhos futuros podem ser desenvolvidos. Entre eles, destacam-se: (I) Realizar a avaliação proposta no presente trabalho em outros aplicativos, com o objetivo de identificar ferramentas que adotam de maneira mais abrangente os métodos ágeis; (II) Identificar de novos critérios relevantes para outros método ágeis, como *Extreme Programming* (XP) e *Lean Software Development*; (III) Elaborar uma pesquisa de opinião com gerentes de projetos para que eles sugiram novos critérios; e (IV) Desenvolver um software com um método multicritério de apoio à decisão para avaliar os softwares automaticamente, ajudando os gerentes a escolher os critérios para fazer a avaliação dos aplicativos e definir suas respectivas importâncias.

Referências

ANDERSON, D.; REINERTSEN, D.; PINTO, A. *Kanban: mudança evolucionária de sucesso para seu negócio de tecnologia*. Washington, DC: Blue Hole Press, 2011.

BECK, K. *et al. Manifesto for agile software development*. [S. l.], 2001. Disponível em: <https://agilemanifesto.org/>. Acesso em: 9 nov. 2023.

BOEG, J. *Kanban em 10 passos*. [S. l.]: C4Media, 2010.

CARVALHO, E. da; MALCHER, P. R.; SANTOS, R. P. dos. A survey research on the use of mobile applications in software project management. In: BRAZILIAN SYMPOSIUM ON SOFTWARE QUALITY, 19., 2020, São Luís. *Proceedings [...]*. New York, NY: Association for Computing Machinery, 2020. p. 1–10. Disponível em: <https://dl.acm.org/doi/10.1145/3439961.3439963>. Acesso em: 9 nov. 2023.

CARVALHO, M. D.; RABECHINI JR, R. *Fundamentos em gestão de projetos: construindo competências para gerenciar projetos*. 5. ed. São Paulo: Atlas, 2019.

CAVALCANTI, F. R. P.; SILVEIRA, J. A. N. *Fundamentos de gestão de projetos: gestão de riscos*. São Paulo: Atlas, 2016.

CODAS, M. M. B. Gerência de projetos: uma reflexão histórica. *Revista de Administração de Empresas*, Rio de Janeiro, v. 27, n. 1, p. 33–37, 1987. Disponível em: <https://www.scielo.br/j/rae/a/Fkq59q8FCfMvqhLPbQpx3pt/>. Acesso em: 9 nov. 2023.

DIGITAL.AI. 15th Annual State Of Agile Report. *Digital.ai*. Raleigh, North Carolina, 2021. Disponível em: <https://info.digital.ai/rs/981-LQX-968/images/SOA15.pdf>. Acesso em: 9 nov. 2023.

DRUMOND, C. O Manifesto Ágil ainda é válido? *Atlassian*, [s. l.], 2021. Disponível em: www.atlassian.com/br/agile/manifesto. Acesso em: 9 nov. 2023.

DRUMOND, C. What is scrum?. *Atlassian*, [s. l.], 2021. Disponível em: www.atlassian.com/agile/scrum. Acesso em: 9 nov. 2023.

FELTRIN, O. A. *Avaliação comparativa de softwares livres para suporte ao desenvolvimento ágil baseado no método ágil scrum*. 2012. Monografia (Especialização em Informática) - Universidade Federal do Paraná, Curitiba, 2012. Disponível em: <https://acervodigital.ufpr.br/handle/1884/38747>. Acesso em: 9 nov. 2023.

FLORES, L. D. S. *Ferramentas de gestão de projetos de desenvolvimento de software: um mapeamento sistemático da literatura*. 2022. Monografia (Bacharelado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2022. Disponível em: <https://lume.ufrgs.br/handle/10183/245266>. Acesso em: 9 nov. 2023.

ISO. *ISO/IEC 25010:2011: systems and software engineering: systems and software quality requirements and evaluation (square): system and software quality models*. Switzerland: ISO, 2011a.

ISO/IEC. *ISO/IEC 25040: 2011: systems and software engineering: systems and software quality requirements and evaluation (square): evaluation process*. Switzerland: ISO, 2011b.

KANBAN UNIVERSITY. *The official guide to the kanban method*. Seattle, 2021. Disponível em: <https://kanban.university/kanban-guide/>. Acesso em: 9 nov. 2023.

MANOLE, M.; AVRAMESCU, M.-S. A comparative analysis of agile project management tools. *Economy Informatics*, [s. l.], v. 17, n. 1, p. 25–31, 2017. Disponível em: <https://economyinformatics.ase.ro/content/EN17/03%20-%20manole,%20avramescu.pdf>. Acesso em: 9 nov. 2023.

PMI. *Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK®)*. 6. ed. Pennsylvania: PMI, 2017.

PRESSMAN, R.; MAXIM, B. *Engenharia de software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.

SCHWABER, K.; SUTHERLAND, J. *The scrum guide: the definitive guide to scrum: the rules of the game*. [S. l.: s. n.], 2020.

SOARES, M. M. *Análise comparativa de ferramentas utilizadas para Kanban*. 2017. Monografia (Bacharelado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017. Disponível em: <https://lume.ufrgs.br/handle/10183/168978>. Acesso em: 9 nov. 2023.

SOMMERVILLE, I. *Engenharia de software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SOMMERVILLE, I. *Software Engineering*. München: Pearson Studium, 2018.

SUTHERLAND, J. *SCRUM*: a arte de fazer o dobro de trabalho na metade do tempo. São Paulo: Leya, 2014.

APÊNDICE A – Funcionalidades dos métodos ágeis

Método	Descrição	Observação
Scrum e Kanban	Cycle Time	
	Taxa de Entrega	
Scrum	Planning Poker	
	Emissão de relatório de atividades do Product Backlog	
	Emissão de relatório de atividades do Sprint Backlog	
	Emissão de relatório/gráfico Sprint Burndown	
	Velocidade da Equipe	
	Gráfico de Burndown	
Kanban	Subdivisões horizontais (raias) ou outro tipo de indicação visual para agrupamento de itens semelhantes	
	Limitação do WIP para cada coluna do Kanban	Apenas Asana não contém
	Lead Time Distribution	
	Gráfico de Fluxo Cumulativo (GFC)	
	Medição do Lead Time	
	Medição do Throughput	

APÊNDICE B – *Script* utilizado para executar a avaliação

Ordem	Descrição	Tempo previsto em minutos
1	Percorrer o sistema analisando se é possível, em uma primeira impressão, compreender a finalidade de cada módulo ou funcionalidade existente.	30
2	Cadastrar um novo projeto no sistema.	2
3	Dentro deste projeto, criar um Product Backlog com no mínimo 3 requisitos, especificando suas prioridades, tempo estimado de execução, breve descrição do que precisa ser feito e responsáveis	10
4	Avaliar os itens 1.2 a 1.4, 2.1 e 3.4 a 3.7	10
5	Criar um Sprint com 5 atividades para o primeiro requisito do Product Backlog, lançar uma data de execução, simulando a execução, para cada atividade, do primeiro Sprint criado.	10
6	Avaliar o item 2.2	2
7	Criar uma lista de impedimentos e adicionar 1 requisito	5
8	Avaliar o item 2.3	2
9	Verificar se existe algum meio de comunicação entre os usuários do sistema (chat, mensagens, etc) e testar seu funcionamento, caso exista.	15
10	Avaliar os itens 3.6 e 3.12	4
11	Verificar se existe a possibilidade de utilizar Planning Poker e testar seu funcionamento, caso exista.	10
12	Avaliar o item 1.1	2
13	Verificar a possibilidade de se criar um projeto novo mantendo o já existente, criar times, separar os itens por time	15
14	Avaliar os itens 3.2 e 3.3	4
15	Verificar se o sistema permite criar diferentes níveis de acesso de usuário.	6
16	Avaliar o item 2.3	2
17	Verificar se o sistema permite a reprodução do quadro Kanban de exemplo, suas subdivisões horizontais e limitação do WIP	10
18	Avaliar os itens 1.5, 1.6 e 2.4	6
19	Verificar se o aplicativo permite a emissão de relatórios	40
20	Avaliar os itens 3.1 a 3.11	20
21	Verificar se o sistema permite exportar e importar um projeto.	8
22	Avaliar os itens 1.7 e 1.8	2
23	Verificar se é citada, em alguma documentação, a possibilidade de se modificar o sistema para ele se comunicar com outros, se for necessário.	10
24	Avaliar o item 2.3	2