



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS FLORIANO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

ROSALICE MARIA PEREIRA LIMA

UM MAPEAMENTO SISTEMÁTICO SOBRE OS MÉTODOS ÁGEIS SCRUM,
KANBAN, SCRUMBAN E EXTREME PROGRAMMING (XP)

FLORIANO

2022

ROSALICE MARIA PEREIRA LIMA

**UM MAPEAMENTO SISTEMÁTICO SOBRE OS MÉTODOS ÁGEIS SCRUM,
KANBAN, SCRUMBAN E EXTREME PROGRAMMING (XP)**

Trabalho de Conclusão de Curso (artigo científico) apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Floriano.

Orientadora: Prof^ª. MSc. Vanessa Veloso Aragão.

FLORIANO

2022

UM MAPEAMENTO SISTEMÁTICO SOBRE OS MÉTODOS ÁGEIS SCRUM, KANBAN, SCRUMBAN E EXTREME PROGRAMMING (XP)

Rosalice Maria Pereira Lima ¹
Prof. MSc. Vanessa Veloso Aragão ²

RESUMO

Este estudo teve como intuito realizar um mapeamento sistemático da literatura sobre os métodos ágeis Scrum, Kanban, Scrumban e XP, visando identificar quais métodos têm sido mais utilizados por empresas de desenvolvimento de software, os métodos ágeis que proporcionam mais produtividade para as equipes de desenvolvimento de software, além disso mostra benefícios dos métodos ágeis em relação os métodos tradicionais. Na metodologia foi realizado o processo de leitura seletiva, analítica e interpretativa a fim de selecionar os estudos mais relevantes para as questões de pesquisa levantadas. A ferramenta *Start* foi utilizada para dar suporte ao processo de mapeamento, otimizando o processo de busca por evidências na literatura científica. Como resultado, esta pesquisa mostra o Scrum como o método ágil mais utilizado por empresas de desenvolvimento de software, que os métodos ágeis são mais vantajosos que os métodos tradicionais devido a adotar uma abordagem leve, sendo iterativo, incremental, dentre outros benefícios. Além disso, foi possível identificar que não é um método específico que gera produtividade em equipes ágeis, e sim fatores que caracterizam a formação de uma equipe, como o tamanho da equipe, o tipo do projeto, o prazo de entrega e dentre outros aspectos.

Palavras-chave: métodos ágeis; produtividade; equipes ágeis; benefícios; desenvolvimento de software.

ABSTRACT

This study aimed to carry out a systematic mapping of the literature on agile methods Scrum, Kanban, Scrumban and XP, in order to identify which methods have been most used by software development companies, the agile methods that provide more productivity for development teams of software, in addition it shows benefits of agile methods in relation to traditional methods. In the methodology, a process of selective, analytical and interpretative reading was carried out in order to select the most relevant studies for the research questions raised. The Start tool was used to support the mapping process, optimizing the process of searching for evidence in the scientific literature. As a result, this research shows Scrum as the agile method most used by software development companies, that agile methods are more advantageous than traditional methods due to adopting a lightweight approach, being iterative, incremental, among other benefits. In addition, it was possible to identify that it is not a specific method that generates productivity in agile teams, but factors that characterize the formation of a team, such as the size of the team, the type of project, the deadline and among other aspects.

Keywords: *agile methods; productivity; agile teams; benefits; software development.*

¹Data de aprovação: 10/01/2023 (data de apresentação do TCC)

¹Graduanda do curso de Tecnologia e Análise de Desenvolvimento de Sistemas do IFPI Floriano. E-mail: caflo2020114tads13@aluno.ifpi.edu.br.

²Professora mestre do curso de Tecnologia e Análise de Desenvolvimento de Sistemas do IFPI Floriano. E-mail: vanessa.veloso@ifpi.edu.br

1 INTRODUÇÃO

Com o surgimento das metodologias ágeis nos anos 90 e a criação do manifesto ágil em 2001, a forma de planejar e construir software em relação aos processos clássicos de desenvolvimento foi repensada. Os métodos tradicionais considerados pesados, difíceis de se manter no cronograma, e com grande quantidade de documentação gerada impulsionaram a mudança de pensamento e a criação de metodologias leves de desenvolvimento de software, as metodologias ágeis (CERQUEIRA et al., 2018).

Visto que as mudanças em projetos acontecem inevitavelmente, e assim as necessidades dos usuários também mudam conforme o tempo, as equipes de desenvolvimento de software precisam ser ágeis para se adaptar a essas mudanças (PRESSMAN, 2011).

Como também, organizações de desenvolvimento de software atuam em ambientes complexos, com mudanças constantes e forte competitividade. Por essa razão, elas têm necessidade de responder de forma ágil às pressões do ambiente (SANTOS; PRADO; CHAIM, 2019). Sendo o *Agile* o tópico decisivo em engenharia de software, garantindo muitos benefícios como reação mais rápida às mudanças, aumento da produtividade da equipe e com uma maior transparência dos projetos (DIEBOLD; SCHMITT; THEOBALD, 2018).

Entre os métodos ágeis que surgiram, tem-se o Scrum, Kanban, Scrumban e Extreme Programming (XP), e a procura pela implantação de Métodos Ágeis nas empresas tem origem em diferentes anseios, como, acelerar a entrega de software, promover a habilidade para tratar mudanças de prioridade, aumentar a produtividade em equipes de desenvolvimento, aperfeiçoar o alinhamento da tecnologia da informação ao negócio, além de melhorar a qualidade do software (AGILE REPORT, 2020).

Conforme indicado pelo Agile Report (2020), 95% das empresas já adotaram algum método ágil, sendo que destas, 37% implantaram no setor de desenvolvimento de software, visando auxiliar ou controlar completamente o Plano de Desenvolvimento de Software estipulado, sendo que em 71% destas teve como principal razão a busca por acelerar a entrega de produtos de software, onde leva-se em consideração para o sucesso da aplicação, principalmente, a escolha de qual método ágil utilizar.

Diante disso, esta pesquisa teve como objetivo principal, apresentar um mapeamento sistemático da literatura sobre os métodos ágeis, Scrum, Kanban, Scrumban, e Extreme Programming (XP) aplicados a projetos de softwares, evidenciando o mais utilizado por empresas de desenvolvimento de software, quais são os benefícios em relação aos métodos tradicionais e quais proporcionam mais produtividade para as equipes.

2 REFERENCIAL TEÓRICO

2.1 MÉTODOS ÁGEIS

No final do século XX e início do século XXI, projetos de desenvolvimento de software enfrentam atrasos em seus cronogramas, aumento no orçamento inicial e falta de qualidade por não seguirem processo algum ou por seguirem um modelo pesado demais para a demanda que se tinha (SOMMERVILLE, 2011).

O modelo seguido era o modelo cascata, também conhecido como *Waterfall*, que é composto por etapas desenvolvidas de forma sequencial, sendo assim, classificado como um modelo linear, pois cada etapa só pode ser iniciada após a finalização da fase anterior, gerando assim dependência entre as fases, além disso, outro fator a ser considerado é que uma má definição dos requisitos impacta diretamente no resultado final do software desenvolvido (TRIGO; BARRETO, 2019).

Com isso, no início de 2001, um grupo de especialistas representantes de metodologias ágeis e simpatizantes, preocupado com o rumo que o desenvolvimento de software estava levando, se reuniram na estação de Ski Snowbird, em Utah, nos Estados Unidos, para discutir maneiras de desenvolver software de uma forma mais leve, rápida e centrada em pessoas (SOMMERVILLE, 2011).

Eles cunharam os termos “Desenvolvimento Ágil de Software” e “Métodos Ágeis” e criaram o manifesto ágil, amplamente difundido como a definição canônica do desenvolvimento ágil. O manifesto foi publicado em 2001, e mais tarde, algumas dessas pessoas formaram a *Agile Alliance* em 2014, uma organização sem fins lucrativos que promove o desenvolvimento ágil (MILANI et al., 2014).

Desse modo, especificamente os métodos ágeis, caracterizam-se por trabalho colaborativo, competências multidisciplinares, envolvimento do cliente, e equipes preferencialmente pequenas. Diferentemente dos métodos tradicionais que mantinham o foco no desenvolvimento individual, pouco envolvimento do cliente, e equipes geralmente grandes (MADADIPOUY, 2015).

2.1.1 Scrum

Dentre os métodos ágeis temos o Scrum, cuja metodologia é utilizada desde 1990, a mesma é usada como um framework estrutural para gerenciamento de trabalhos e produtos complexos. O Scrum deixa explícito a eficácia relativa de suas práticas de gerenciamento de produtos, processos e técnicas de trabalho, de modo que o produto seja continuamente melhorado, a equipe e o ambiente de trabalho. Dado a conceituação do Scrum, destaca-se que o método possui três pilares importantes, a transparência, a inspeção e a adaptação (SCHWABER; SUTHERLAND, 2017).

O método utiliza as abordagens iterativa e incremental, de forma a possibilitar que a equipe de desenvolvimento faça modificações e alterações no projeto visando melhorar o produto durante o ato da sua criação (CARVALHO et al., 2020).

Além disso, as atividades no desenvolvimento do software são realizadas em "sprints", que é o termo dado para um pequeno período de tempo onde um conjunto de tarefas é realizada. O número de sprints necessários para cada atividade metodológica pode ter variações dependendo do tamanho e da complexidade do produto em desenvolvimento (PRESSMAN, 2011).

Para entender como funciona o modelo Scrum, é necessário elaborar um panorama dos papéis desempenhados por cada integrante do time, sendo que dentro do time Scrum, não há hierarquias, o time consiste em um product owner, um scrum master e uma equipe desenvolvedores (SCHWABER; SUTHERLAND, 2020).

O product owner, é sumariamente, o dono do produto, ele é o responsável por expandir o valor do item a ser desenvolvido e também do trabalho realizado pelo time de desenvolvedores (SCHWABER; SUTHERLAND, 2020).

Já o scrum master, é o responsável por garantir que as regras e boas práticas do Scrum sejam utilizadas durante o andamento do projeto, respeitando seus valores (SCHWABER; SUTHERLAND, 2020).

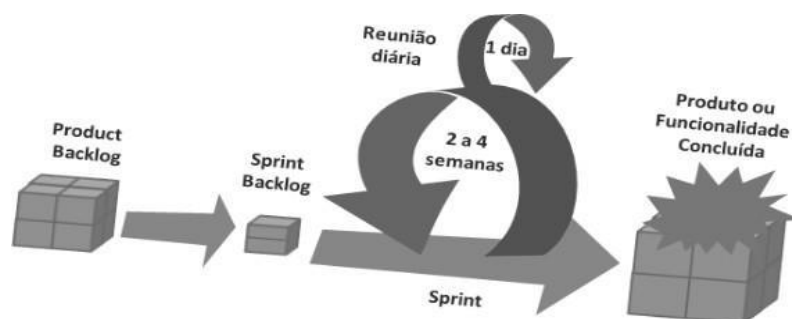
O time Scrum conta também com uma equipe de desenvolvedores, que são responsáveis pelo desenvolvimento de um software, tangendo no seu conjunto de responsabilidades a criação de um plano para a sprint, o sprint backlog e a introdução gradual de qualidade, aderindo a uma definição de pronto a cada incremento realizado no produto (SCHWABER; SUTHERLAND, 2020).

Conforme os conceitos sobre os papéis do time scrum, Oliveira e Muniz (2015) afirmam que os mesmos se associa com os elementos contidos no Scrum como, eventos e atributos, sendo o ponto principal o Backlog do Produto, que são os registros dos requisitos

do produto, onde são coletadas as funcionalidades e necessidades do negócio, em ordem prioritária, por meio de reuniões com os stakeholders.

A Figura 1. mostra o processo do Scrum, onde os projetos são divididos em ciclos chamados de sprints, que ocorrerem entre uma semana e um mês, precedido por uma reunião de planejamento, onde tarefas são identificadas e um compromisso para o objetivo do sprint é definido, seguido por uma reunião de revisão ou de retrospectiva, onde o progresso é revisto e lições para os próximos sprints são identificadas. Todas as sprints são postas em productbacklog. O product backlog é mantido pelo product owner e é uma lista de requisitos que tipicamente vêm do cliente, depois de observar as suas necessidades. O sprint representa um time box dentro do qual um conjunto de atividades deve ser executado. Ele apresenta papéis principais que são: o product owner, o scrum master e o scrum team.

Figura 1 - Esquema geral do processo scrum



Fonte: Bastos e Bastos (2021).

2.1.2 Kanban

Segundo Tubino (2009), o sistema Kanban foi desenvolvido na década de 60 por Taiichi Ohno, então gerente de um setor da montadora Toyota no Japão, o mesmo foi desenvolvido com base na dinâmica de operação dos supermercados e no seu sistema de reposição de produtos nas prateleiras.

A palavra Kanban no japonês apresenta alguns significados como por exemplo, cartão, painel ou símbolo, de forma resumida o kanban tem por finalidade controlar a produção e diminuir os estoques usados no processo de produção, fabricando em poucas quantidades para não haver prejuízo e tudo sair conforme o planejado dentro do tempo previsto (SILVA; ANASTACIO, 2019).

Este método está relacionado à utilização de “cartão” ou “sinalização” com cores diferentes para indicar o andamento dos fluxos de produção desde o seu ponto inicial até o final, indicando o status de determinadas tarefas, como finalizado, em andamento ou não executado (SOUSA et al., 2017).

O Kanban é uma ferramenta utilizada conjuntamente com o Just In Time, que é um termo inglês, que significa literalmente “na hora certa” ou “momento certo” (SILVEIRA, 2018) para o controle de produção, uma técnica, onde cartões controlam o estoque e a produção de forma visual.

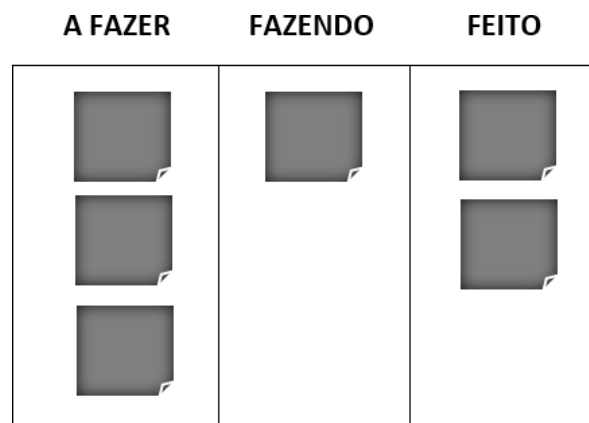
No *Just In Time*, mais do que uma técnica de gestão de produção, existe uma filosofia completa, baseada em conceitos de produtividade, qualidade, gestão de materiais, projeto do produto, gestão de pessoas. Os dois conceitos operam se complementando (SILVEIRA, 2018).

Essa filosofia, conforme esclarecem é traduzida em melhoria contínua e eliminação de

desperdício, proporcionando produção com custo efetivo e entrega somente do que é necessário, com qualidade, na quantidade, tempo e lugar certo, utilizando o mínimo de recursos materiais, humanos e financeiros (SILVA; ANASTÁCIO, 2019).

De forma simplificada, esse modelo consiste em utilizar o quadro Kanban, que pode ser feito em um quadro branco, parede ou folha onde são colados cartões que demonstram o fluxo do trabalho. À medida que o trabalho vai sendo realizado, os cartões avançam dentre os estágios pré-estabelecidos: a fazer, em andamento e finalizado. Na Figura 2 conseguimos visualizar esta ideia.

Figura 2 - Quadro Kanban



Fonte: Elaborado pela autora (2022).

2.1.3 Scrumban

O Scrumban combina as melhores características do Scrum e do Kanban, tendo como objetivo principal responder às mudanças do usuário, utiliza-se do planejamento e da iteração do Scrum permitindo alterações em suas sprints e herdando a visualização de fluxos conforme quadro Kanban (PLENGVITTAYA; SANPOTE, 2018).

O Scrumban é um modelo híbrido que torna o Scrum mais enxuto e flexível para mudanças rápidas, utilização do planejamento e revisão de sprint, combinando com partes do modelo Kanban que utiliza os quadros de fluxo de trabalho, WIP, regras de equipe, etc (LUTFIANI, et al., 2020).

Segundo o estudo empírico feito por Alqudah e Razali (2018), ao realizar um projeto de desenvolvimento de produto ou serviço, é raro o método do Scrum ou do Kanban se adequarem perfeitamente aos critérios do projeto em andamento. O Scrumban apoia os membros da equipe ágil a serem criativos no desenvolvimento de novos métodos para atender às suas necessidades. Não existem práticas específicas para o Scrumban, mas os membros da equipe ágil precisam entender quais práticas do Scrum e do Kanban agregam valor e escolhem as práticas apropriadas de acordo. O Scrumban, quando utilizado por alguém com experiência nos dois métodos, permite definir a duração dos sprints e limitar o *Work-In-Process* (WIP), que significa limitar a quantidade de tarefas em andamento.

No Scrumban também são realizadas reuniões de planejamento, ao contrário do Scrum que têm um planejamento curto. Revisões, retrospectivas e reuniões diárias são sempre realizadas, fornecendo feedback diretamente do proprietário do produto ou de quem tem um relacionamento com a equipe, como gerentes e clientes (LUTFIANI et al., 2020).

Algumas vantagens do Scrumban, são, o curto período de entrega, ou seja, a diminuição do prazo de execução, a minimização de desperdício, ou seja, tudo o que não agrega valor ao cliente, a melhoria de processo, adicionando alguns valores do Scrum, e

Kanban, como e quando necessário (SOTILE, 2018). Este método é usado principalmente para projetos em andamento, para equipes que veem o Scrum como um método muito rigoroso e precisam da flexibilidade do Kanban (KEUP, 2020).

2.1.4 Extreme programming (XP)

Segundo Wildt et al. (2015), O eXtreme Programming é uma metodologia ágil de desenvolvimento de software voltada para times de pequeno a médio porte, onde os requisitos são vagos e mudam frequentemente. Sendo um método de desenvolvimento de software com foco na agilidade de equipes e na satisfação do cliente (WILDT ET AL. 2015).

O XP possui três características importantes, que são, a abordagem incremental, o feedback constante, e o encorajamento da comunicação entre as pessoas, clientes e equipe. O XP também é uma metodologia de desenvolvimento de software que dá a preferência ao desenvolvimento orientado a objetos, sendo mais adequada para pequenas e médias equipes, além de ser uma metodologia que utiliza o modelo incremental, ou seja, enquanto o software é utilizado, novas melhorias são implementadas (SBROCCO; MACEDO, 2012).

O XP sempre está buscando fundamentar suas práticas em um conjunto de valores bem definidos, além de se empenha em assegurar que a equipe se concentre em desenvolver primeiramente o que é realmente necessário para o funcionamento do projeto, ao invés de colocar no projeto várias funcionalidades que o cliente não precisa no momento, e podem ser inseridas futuramente (SBROCCO; MACEDO, 2012).

O XP possui doze práticas sugeridas na metodologia, no Quadro 1 conseguimos visualizar esta ideia.

Quadro 1 - Práticas XP

Práticas	Como funcionam
Cliente sempre disponível	Todas as atividades, a partir da segunda iteração, são realizadas em pares
Metáforas no projeto	Uma metáfora pode ser usada para descrever um requisito
Jogo de planejamento	Em que a cada iteração, a equipe tem um tempo para priorizar e planejar novamente atividades
Entregas pequenas e rápidas	Faz com que as equipes vão entregando partes pequenas a cada vez, de forma rápida
Teste de aceitação	O cliente informa se a atividade é válida e está aprovada ou não
Refatoração	Com essa prática algumas atividades precisam ser melhoradas em iterações posteriores;
Integração contínua	Algumas atividades, no decorrer do processo, necessitam da integração entre outras atividades
Propriedade coletiva	Todas as tarefas devem ser executadas por todos os membros, trocando atividades ou se ajudando
Design simples	Tarefas devem ser efetuadas prevendo a simplicidade, sob pena da equipe perder tempo de forma desnecessária
Ritmo sustentável	Essa prática está tocante com a restrição de todos os membros do time estarem no local programado ao final de cada integração, sob pena de perder tempo na próxima iteração
Padrões de código	Onde algumas atividades, durante o processo, necessitam da integração entre outras atividades

Fonte: Schoeffel (2021).

2.2 EQUIPES ÁGEIS E PRODUTIVIDADE

A produtividade pode ser definida como a quantidade de trabalho feita por unidade de tempo, assim, no desenvolvimento de software com tarefas que dependem de esforços colaborativos e criativos, medir tempo é relativamente simples. Entretanto, definir produtividade não é uma atividade fácil, uma vez que desafia pesquisadores, projetistas e desenvolvedores a entender e melhorar seus próprios processos, de forma a buscar melhor rendimento de suas atividades, introduzindo novas ferramentas, ambientes, estruturas, um contexto específico, além de práticas organizacionais para a formação de equipes ágeis de desenvolvimento de software (OLIVEIRA et al., 2020).

A produtividade da equipe é um fator determinante para o sucesso do desenvolvimento de software, quando bem empregado, características como, o tipo do projeto, o tamanho da equipe, o prazo de entrega em conjunto com o uso os métodos ágeis resultam em projetos com baixo risco, e alta qualidade, que seguem um estilo de desenvolvimento iterativo e incremental, onde, de forma colaborativa, as equipes se auto-organizam e se ajustam dinamicamente às mudanças nos requisitos do cliente, tornando se equipes produtivas (CALDEIRA et al., 2019).

A experiência dos integrantes da equipe também influencia a produtividade no desenvolvimento de software de maneira positiva, além de que, o relacionamento entre as habilidades e competências, a motivação dos envolvidos, o trabalho colaborativo, e a facilidade de comunicação entre os membros, são fatores importantes para o aumento da produtividade (MOTA, 2021).

3 METODOLOGIA

3.1 CARACTERIZAÇÃO DA PESQUISA

O Mapeamento Sistemático da Literatura ou revisão de escopo é utilizado quando não é necessário responder com profundidade questões específicas, podendo ser visto como uma revisão mais extensa dos estudos primários disponíveis na literatura, onde seu principal objetivo é fornecer uma visão geral de uma área de pesquisa e identificar a quantidade de estudos, e os resultados disponíveis nessa área, onde para obtenção desse objetivo o mapeamento sistemático deve ser guiado por meio de questões de pesquisa, que devem ser respondidas ao final do processo de revisão da literatura (MOHER; SHEKELLE, 2015).

Deste modo, a pesquisa realizada configura-se como um Mapeamento Sistemático da Literatura, sendo escolhido por ser um processo metodológico cuidadosamente controlado por um protocolo de investigação formal que contém, itens como, questões de pesquisa, estratégia de busca, critérios para inclusão e exclusão de estudos e estratégias para seleção, extração e sumarização dos dados, fornecendo consistência, veracidade e robustez na apresentação dos seus resultados.

3.2 FERRAMENTA START

A ferramenta utilizada para a realização do Mapeamento Sistemático foi o *Start*, que é um programa desenvolvido pelo Laboratório de Pesquisa em Engenharia de Software da Universidade Federal de São Carlos (LaPES-UFSCAR) no Brasil. O Start é de uso gratuito e auxilia nas pesquisas e desenvolvimento de estados da arte, revisões sistemáticas da literatura e revisões de escopo (START, 2020).

A ferramenta possibilita uma sistematização no processo de revisão de literatura, desde a inclusão das referências nas bases, com um protocolo, execução e sumarização dos resultados

otimizando o trabalho que seria mais prolongado sem uso da mesma (PIMENTEL; SALES JUNIOR, 2021).

No protocolo, são definidos os detalhes de execução planejados para o mapeamento sistemático, como o objetivo da pesquisa, as palavras-chave, questões de pesquisa, critérios de inclusão e exclusão, a estratégia para avaliação de qualidade dos artigos. A execução que é onde os dados deverão ser extraídos, levando em consideração os critérios de inclusão e exclusão, e a sumarização, que é a forma como os estudos deverão ser sintetizados, onde os resultados da execução podem ser visualizados sob a forma de gráficos (FABBRI et al., 2016).

3.3 QUESTÕES DE PESQUISA

Considerando o objetivo da pesquisa apresentada, foram definidas as seguintes questões de pesquisa: 1) Quais métodos ágeis têm sido mais utilizados por empresas de desenvolvimento de software? 2) Quais os benefícios dos métodos ágeis em relação aos métodos tradicionais? 3) Quais métodos proporcionam mais produtividade para as equipes de desenvolvimento?

3.4 ESTRATÉGIA DE BUSCA

Optou-se pela busca em duas bases de dados científicas (IEEE Xplore e Web of Science), os trabalhos foram publicados entre o período de 2019 e 2022, nos idiomas inglês e português. Utilizou-se a *string* de busca ((*agile methods* OR *most used*) OR (*software development companies* OR *benefits*) OR (*produtividade* AND *agile teams*) OR (*scrumban* OR *Extreme Programming* AND *Kanban* AND *Scrumban*)). A string foi usada em ambas as bases e construída com palavras-chave da pesquisa.

3.5 CRITÉRIOS DE INCLUSÃO E EXCLUSÃO

Os critérios de inclusão e de exclusão são importantes para filtrar estudos não relevantes para a aquisição de respostas às questões de pesquisa. O critério de inclusão (CI) e de exclusão (CE) considerados neste trabalho apresentam-se no Quadro 2.

Quadro 2 - Critérios de inclusão e Exclusão

Critérios de Inclusão (CI)	Critérios de Exclusão (CE)
CI1: Estudos que apresentem a aplicação de métodos ágeis em projetos de software	CE1: Trabalhos que não tenham relação com o tema da pesquisa
CI2: Título e resumo de acordo com o que contenham as palavras chaves	CE2: Trabalhos Incompletos
CI3: Publicações que apresentam comparativos entre os métodos ágeis	CE3: Trabalhos não alinhados com o objetivo geral da pesquisa
CI4: Artigos publicados nos últimos três anos	CE4: Trabalhos publicados em outras línguas que não sejam português e inglês.

Fonte: Elaborado pela autora (2022).

3.6 PROTOCOLO DE BUSCA

Em princípio foi realizada a configuração da ferramenta Start para a execução do Mapeamento, com o preenchimento de um protocolo, no qual são definidos os objetivos da pesquisa, as palavras-chave, as questões de pesquisa e os critérios de inclusão e exclusão que foram utilizados.

Assim, foi realizada a busca manual de trabalhos nas fontes de dados IEEE Xplore e *Web of Science*, utilizando as strings de busca apresentadas anteriormente, compostas por meio da combinação de palavras-chave relacionadas às questões levantadas, por meio dos mecanismos de busca avançada. A partir disso, 100 potenciais estudos foram exportados, sendo 50 artigos da base de dados IEEE e 50 artigos do *Web of Science*, deste modo, os trabalhos retornados pela busca foram exportados para a ferramenta Start no formato Bibtex.

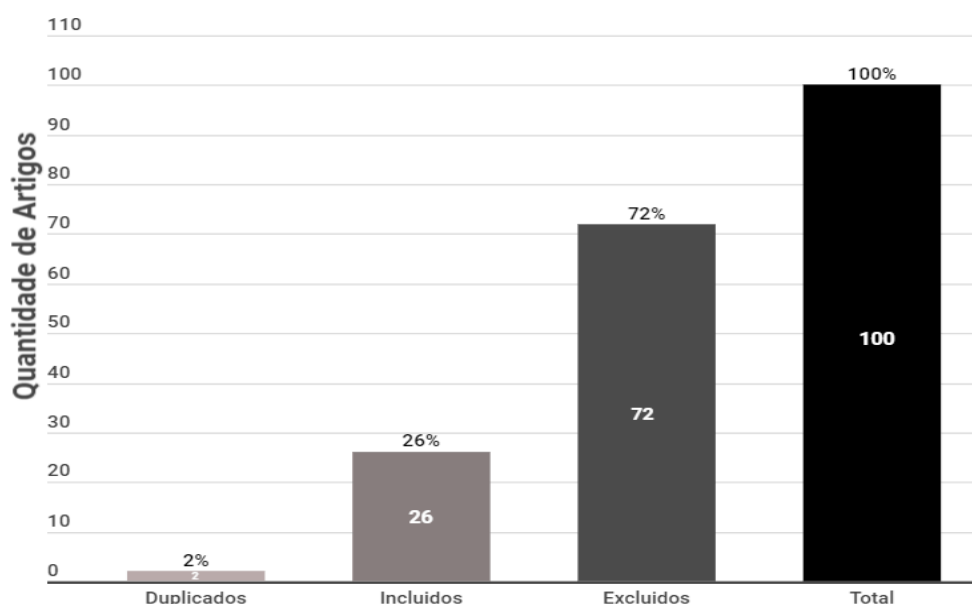
3.7 LEITURA E SELEÇÃO DOS DADOS

Sobre a seleção de leitura das obras para a análise dos dados, primeiramente foi realizada a leitura seletiva dos estudos, onde a leitura das obras foi feita por completo, a fim de selecionar as informações de interesse, selecionando dados relevantes que poderão contribuir para o andamento da pesquisa. Após a leitura seletiva, foi realizada a leitura analítica, que é uma leitura compassada, que tem por objetivo a absorção total do conteúdo, ou seja, posições e argumentos que permitirão o entendimento da pesquisa analisada. E por fim, a leitura interpretativa, sendo mais completa, onde houve o aprofundado das ideias principais, buscando identificar o que o autor afirma e quais informações ele oferece como fundamento, correlacionado às afirmações do autor com os problemas em questão (GIL, 2007).

Nessa seleção, o título, o resumo e as palavras-chave dos 100 estudos foram analisados, desses, 2 trabalhos que eram duplicados foram descartados, 72 trabalhos foram excluídos de acordo com os critérios de exclusão e 26 trabalhos foram incluídos para a leitura seletiva.

Desse modo, foi realizada a leitura seletiva, onde os trabalhos foram lidos por completo, verificando quais melhores poderiam responder às questões de pesquisa levantadas. A Figura 3 mostra o resultado da busca pela ferramenta Start, encontrou-se (2%) artigos duplicados, (72%) artigos excluídos e (26%) artigos incluídos na fase de pré-seleção.

Figura 3 - Resultado da seleção preliminar



Fonte: Pesquisa da autora (2022).

Após a leitura seletiva, foi realizada a leitura analítica do texto completo desses 26 estudos, sendo uma leitura compassada, que permitiu o entendimento das pesquisas analisadas.

Assim a aprovação para extração dos resultados foi determinada novamente com base nos critérios de inclusão e exclusão. E desses 26 artigos lidos por completo, 17 foram considerados relevantes para a pesquisa e 9 foram excluídos. Assim, os 17 artigos selecionados foram utilizados para a extração dos dados e análise, na qual foi realizada assim, a leitura interpretativa dos mesmos, correlacionado às afirmações do autor com os problemas em questão.

Com o novo conjunto de estudos obtido, foram extraídos os dados necessários para responder às questões de pesquisa. Os dados foram extraídos de acordo com as seguintes informações: título do artigo, autores, ano de publicação, fonte, palavras chave do artigo, tipo de pesquisa, metodologia utilizada, resumo e questões de pesquisa a serem respondidas.

4 RESULTADOS

Nesta seção são apresentados os resultados obtidos a partir dos 17 artigos selecionados para os estudos finais (extração de dados e análise) relativos aos critérios de inclusão (CI) desta pesquisa. O Quadro 3 exibe os artigos selecionados com base nesses critérios.

Quadro 3 - Artigos da análise final da pesquisa

ID	Título do Artigo	Autor	Fonte de Dados	Ano	CI
1	Real World Scrum A Grounded Theory of Variations in Practice	(MASOOD; HODA; BLINCOE, 2020)	IEEE	2022	CI2 e CI4
2	Conceptual Framework for Auditing Agile Projects	(MKOBA; MARNEWICK, 2020)	IEEE	2020	CI1, CI2 e CI4
3	Scrum Watch: a tool for monitoring the performance of Scrum-based workteams	(VEGA et al., 2022)	Web of Science	2022	CI1, CI2 e CI4
4	Analytic Hierarchy Process Based Prioritisation and Taxonomy of SuccessFactors for Scaling Agile Methods in Global Software Development	(SHAMEEM et al., 2020)	Web of Science	2020	CI1 e CI4
5	Implementing ISO/IEC 29110 to reinforce four very small entities of Mexico under an agile approach	(MUÑOZ; MEJIA; LAPORTE, 2020)	Web of Science	2019	CI1 e CI4
6	Social loafing prevention in agile software development teams using team expectations agreements	(FRONZA; WANG, 2021)	Web of Science	2021	CI2 e CI4
7	Satisfaction and its correlates in agile software development	(KROPP et al., 2020)	Web of Science	2020	CI1, CI2 e CI4

8	Collaborative experience between scientific software projects using Agile Scrum development	(BAXTER et al., 2022)	Web of Science	2022	CI1 e CI4
9	A Genetic Algorithm-Based Approach to Support Forming Multiple Scrum Project Teams	(COSTA et al., 2022)	IEEE	2022	CI1, CI2 e CI4
10	Impact of software development processes on the outcomes of student computing projects: A tale of two universities	(WŁODARSKI; PONISZEWSKA-MARANDA; FALLERI, 2022)	Web of Science	2022	CI2 e CI4
11	An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review	(FERNANDEZ-DIEGO et al., 2020)	IEEE	2020	CI1, CI2, CI3 e CI4
12	Tailoring Agile-Based Software Development Processes	(AKBAR, 2019)	IEEE	2019	CI2 e CI4
13	An empirical study on changing leadership in agile teams	(SPIEGLER; HEINECKE; WAGNER, 2021)	Web of Science	2021	CI2 e CI4
14	Measuring and Improving Agile Processes in a Small-Size Software Development Company	(CHORAS et al., 2020)	IEEE	2020	CI1 e CI4
15	Team Formation in Software Engineering: A Systematic Mapping Study	(COSTA et al., 2020)	IEEE	2020	CI2 e CI4
16	Agile Software Development Using Cloud Computing: A Case Study	(YOUNAS et al., 2020)	IEEE	2020	CI1 e CI4
17	Translation of Activities in a Global Virtual Teams Software Development: Agile vs. Waterfall	(ADELAKUN; IYAMU, 2021)	Web of Science	2021	CI2 e CI4

Fonte: Elaborado pela autora (2022).

4.1 MÉTODOS ÁGEIS MAIS UTILIZADOS POR EMPRESAS DE DESENVOLVIMENTO DE SOFTWARE

Os resultados da pesquisa mostraram que o método ágil mais utilizado por empresas de

desenvolvimento de software é o Scrum, devido a aspectos que o tornam consideravelmente melhor, como uma abordagem leve, sendo iterativo, incremental, além de possuir papéis claramente definidos, e uma fácil implementação, relacionado aos demais métodos estudados na pesquisa, entre outros aspectos apresentados no Quadro 4.

Quadro 4 - Aspectos motivadores para implantação do Scrum

Aspectos motivadores para implantação do Scrum	Estudos de referência
Abordagem leve, iterativo, incremental e popular	O trabalho relata que o Scrum é iterativo e incremental com enfoque em práticas de gerenciamento de projetos, o Scrum possui abordagem consideravelmente leve, além de ser o método ágil mais popular e usado por empresas de desenvolvimento de software (MASOOD; HODA; BLINCOE, 2020).
Revisões de Código	Entre os métodos mais usados por empresas de desenvolvimento de softwares tem-se o Scrum, Lean e XP, onde o Scrum se destaca com 54% de uso em 2018, Lean aparece em segundo com 2%, e XP com somente 1%, onde no Scrum as revisões de código se destacam por ajudar a melhorar a qualidade e o compartilhamento de conhecimento entre os desenvolvedores dos projetos (SHAMEEM et al., 2020).
Papéis claramente definidos	Uma pesquisa feita pela Version One em 2019 concluiu que 72% das empresas de desenvolvimento de software pesquisadas usavam o Scrum dentre os métodos ágeis, ou um híbrido que inclui Scrum, pois os papéis e expectativas no Scrum são claramente definidos em relação a outros métodos, permitindo que os indivíduos executem suas tarefas com eficiência com a existência de três papéis bem definidos, o Product Owner, o Time de Desenvolvimento e o Scrum Master (VEGA et al., 2022).
Equipes auto-organizadas, eventos	O Scrum instrui equipes auto-organizadas para dividir seu trabalho em ciclos de 1 a 4 semanas que são denominados 'Sprints', nos quais visam construir softwares funcionais que atendam as necessidades do cliente, o Scrum possui também seis eventos especificamente, que são, a reunião de visão do projeto, o planejamento do sprint, o sprint, as reuniões diárias, a revisão do sprint e a retrospectiva do sprint, que o tornam um método que apresenta muitos bons resultados em empresas de desenvolvimento de software. como também em qualquer organização (MUÑOZ; MEJIA; LAPORTE, 2020).
Alta qualidade do trabalho final	Um elemento-chave para a abordagem Scrum que o torna muito utilizado por empresas de software é que cada sprint leva à conclusão das tarefas identificadas durante o planejamento do sprint, mesmo que os requisitos mudem no meio do sprint, repercutindo na alta qualidade do trabalho final, reduzindo riscos, custos e erros (BAXTER et al., 2022).
Combinações com outros métodos	O trabalho relata que o Scrum é identificado em 67,12% dos estudos primários da pesquisa, como método ágil mais utilizado. Além disso, os demais usam métodos em combinação com o Scrum para melhor eficiência nos processos (FERNANDEZ-DIEGO et al., 2020).

Evolução de papéis	Atualmente, o método ágil mais conhecido e utilizado é o Scrum, onde, desde a sua aplicação em uma empresa de software, a descrição dos papéis evoluem de um foco específico para cuidar da dinâmica da equipe, formando-se Scrum Masters, que é o responsável por ajudar a equipe a compreender inteiramente a metodologia, seus valores, princípios e práticas (SPIEGLER; HEINECKE; WAGNER, 2021).
Eficiência e qualidade	Os autores deste estudo afirmam que o Scrum pode se mostrar uma boa alternativa para melhoria de processos em uma organização de desenvolvimento de software com recursos muito limitados, devido ser um método eficiente que obtém também bons resultados com menos gente, tempo e recursos (CHORAS et al., 2020).
Fácil implementação	Dentre muitas práticas ágeis, como Scrum, Kanban, eXtreme Programming, Scrumban e etc. a mais popular e mais adotada é Scrum, que provavelmente é o mais adotado porque é relativamente fácil de implementar (ADELAKUN; IYAMU, 2021).

Fonte: Elaborado pela autora (2022).

4.2 BENEFÍCIOS DOS MÉTODOS ÁGEIS EM RELAÇÃO OS MÉTODOS TRADICIONAIS

Observou-se após a análise dos dados coletados no mapeamento sistemático benefícios dos métodos ágeis em relação aos métodos tradicionais, como, a melhoria nas taxas de sucesso dos projetos de TI com o uso de métodos ágeis, equipes auto-organizadas, equipes mais produtivas, pequenos ciclos de desenvolvimento e as melhorias contínuas ao longo dos processos tornam os métodos ágeis mais vantajosos em relação aos métodos tradicionais, onde a utilização daqueles foi citada em estudos visando atender o objetivo de desenvolver um software de forma rápida e incremental, sendo que os métodos tradicionais são considerados lentos no processo de desenvolvimento, apresentando dificuldades em cumprir o cronograma do projeto e os prazos de entrega. No quadro 5 são apresentados mais alguns desses benefícios.

Quadro 5 - Benefícios dos métodos ágeis

Benefícios	Estudos de referência
Melhoria nas taxas de sucesso dos projetos de TI, equipes auto-organizadas, software de alta qualidade	O estudo de Mkoba e Marnewick (2020), mostra que 71% das organizações mundiais de desenvolvimento de software estão adotando o gerenciamento ágil de projetos, gerando melhoras nas taxas de sucesso dos projetos de TI. O ágil está promovendo o conceito de equipes auto-organizadas que organizam seu trabalho por conta própria, além disso, os métodos ágeis formados por princípios, entregam software de alta qualidade, aumentando o uso de ágil nas indústrias de software e TI que vem em contrapartida aos métodos tradicionais, tornando os métodos ágeis mais vantajosos (SPIEGLER; HEINECKE; WAGNER, 2021).
Mobilidade, flexibilidade, habilidades sociais e potencial	Alguns dos benefícios na aplicação dos métodos ágeis por empresas de desenvolvimento de software é a mobilidade que o ágil oferece, a flexibilidade nas mudanças ao longo dos processos, habilidades sociais e o potencial em um curto espaço de tempo para entregar novos produtos e serviços de alto valor ao mercado, características que os métodos ágeis não apresentam (VEGA et al., 2022).

Colaboração do cliente	Nos métodos ágeis, as atividades de desenvolvimento e teste são executadas simultaneamente com a participação do cliente no decorrer do projeto (SHAMEEM et al., 2020), diferentemente da abordagem tradicional, onde problemas podem ser agravados, pois as partes interessadas e os desenvolvedores estabelecem requisitos de software no início do projeto e depois desenvolvem para atender a esses requisitos, sendo que se os requisitos mudam durante o período de desenvolvimento, o software estará obsoleto quando entregue (BAXTER et al., 2022).
Rápido desenvolvimento de software	Os métodos ágeis visam atender o objetivo de desenvolver um software de forma rápida, de forma incremental, já os métodos tradicionais são lentos no desenvolvimento de software pois apresentam dificuldades em cumprir o cronograma do projeto e os prazos de entrega (MUÑOZ; MEJIA; LAPORTE, 2020; ADELAKUN; IYAMU, 2021).
Equipes produtivas	Os métodos ágeis se destacam dos métodos tradicionais em questão de benefícios, pois, no mesmo, a atividade de desenvolvimento é produtiva e bem-sucedida a partir de que os membros da equipe desenvolvem entendimentos comuns sobre diversos aspectos, como objetivos, requisitos e responsabilidades individuais (FRONZA; WANG, 2021).
Pequenos ciclos de desenvolvimento e melhorias contínuas	Encontrou evidências de que os praticantes de Agile estavam mais satisfeitos do que os que ainda utilizavam métodos tradicionais, vê-se uma taxa de satisfação muito elevada, tanto para as empresas como para clientes e profissionais individuais (KROPP et al., 2020). Devido a pequenos ciclos de desenvolvimento, a melhorias contínuas de design, na entrega contínua e no feedback dos clientes, características dos métodos ágeis (YOUNAS et al., 2020).
Planejamento gradual	Com o uso de métodos ágeis o planejamento é gradual, sendo mais fácil mudar o processo para refletir as mudanças nas necessidades dos clientes, diferentemente dos métodos tradicionais, que são engessados, não permitindo mudanças ao longo da execução (COSTA et al., 2020).
Cronograma planejado	Hoje, a maioria das pequenas e médias empresas preferem abordagens ágeis do que as tradicionais para produzir software, pelo esforço para atender o cronograma solicitado pelo mercado, pois o cronograma ágil é bem planejado com maior índice de confiabilidade (MUÑOZ; MEJIA; LAPORTE, 2020).
Produção reduzida de documentação	As metodologias ágeis diferem das abordagens tradicionais de desenvolvimento de software de forma a produzir menos documentação do projeto, onde nos métodos tradicionais os projetos perdem a transparência devido a esse excesso de documentações (SHAMEEM et al., 2020).

Fonte: Autoria própria (2022).

4.1 MÉTODOS ÁGEIS QUE PROPORCIONAM MAIS PRODUTIVIDADE PARA AS EQUIPES DE DESENVOLVIMENTO DE SOFTWARE

Os resultados da pesquisa mostram que não é um método específico que proporciona mais produtividade para as equipes de desenvolvimento de software, e sim que, cada modelo

de desenvolvimento de software valoriza o trabalho em equipe de forma diferente, sendo que, o trabalho em equipe é uma característica central do desenvolvimento ágil, ligado a práticas organizacionais, assim, fatores como o estabelecimento de um terreno comum entre todos os membros, regras e objetivos que todos concordam em respeitar, são fundamentais para uma cooperação bem sucedida em qualquer tipo de projeto para o aumento da produtividade. (FRONZA; WANG, 2021; SHAMEEM et al., 2020).

Assim, é proposto que o trabalho em equipe eficaz não pode acontecer sem estabelecer um levantamento de requisitos, junto ao cliente, sendo a primeira fase no ciclo de desenvolvimento, onde são definidas as funcionalidades e o escopo do projeto, sendo fundamental para uma cooperação bem sucedida, onde o método deve ser escolhido em um determinado contexto específico, considerando fatores como, o tamanho do projeto, o tipo do projeto, o prazo de entrega, escopo, orçamento disponível, tamanho da equipe e outras características que determinaram o sucesso (FRONZA; WANG, 2021).

Todas as práticas ágeis têm princípios semelhantes que favorecem o progresso incremental rápido e a colaboração entre os membros da equipe de forma produtiva (ADELAKUN; IYAMU, 2021), assim, equipes ágeis que priorizam a comunicação sobre a documentação, onde eles podem ser mais produtivos, exigem uma boa liderança para gerenciar e facilitar a interação auxiliando os membros durante conflitos, que consequentemente fazem parte do processo (COSTA et al., 2020; YOUNAS et al., 2020).

Além do conhecimento do desenvolvedor, também utiliza-se suas habilidades para a formação de equipes produtivas, como, os serviços que o sistema ou software oferecerá e dispõe sobre as restrições à operação dele, sobre qual método proporcionará maior produtividade naquele trabalho, sendo também, que a forma como o projeto é concebido e executado é essencial para sua execução de forma produtiva (WŁODARSKI; PONISZEWSKA-MARAÑDA; FALLERI, 2022).

As equipes ágeis priorizam a comunicação sobre a documentação, assim, eles podem ser mais colaborativos na formação da documentação, se tornando um processo mais eficiente e produtivo, outros fatores que tornam a execução do projeto produtiva é a formação de uma equipe, a partir de informações históricas, de sucesso em projetos passados, do tempo de experiência com linguagens de programação, do número de projetos em que o desenvolvedor trabalhou anteriormente e o status dos projetos (COSTA et al., 2022).

Por exemplo, não é suficiente saber se um desenvolvedor trabalhou com Java, tem-se que saber especificamente se foi para desenvolvimento Web, Desktop ou Mobile, além do conhecimento do desenvolvedor, também deve-se utilizar de suas habilidades para a formação de equipes, somente assim são formadas equipes que geram produtividade na execução das atividades (COSTA et al., 2022).

Há também uma maior satisfação do desenvolvedor, com o produto, com a colaboração do cliente no levantamento de requisitos, e no uso de métodos ágeis, onde a moral da equipe, a produtividade da equipe e o desenvolvimento de pessoas são citados como os principais benefícios da execução de práticas ágeis independente da metodologia utilizada (KROPP et al., 2020).

5 CONCLUSÃO

Com a execução desse estudo foi possível identificar que dentre os métodos ágeis abordados na pesquisa, o método ágil Scrum é o mais utilizado por empresas de desenvolvimento de software, devido a inúmeras vantagens que se sobressaem em relação aos demais métodos estudados na pesquisa como, fácil implementação, abordagem leve, além de possuir papéis claramente definidos, sendo iterativo e incremental.

A pesquisa identificou também benefícios dos métodos ágeis em relação aos métodos tradicionais, como a melhoria nas taxas de sucesso dos projetos de TI, a formação de equipes auto-organizadas, o desenvolvimento de softwares de alta qualidade e a colaboração do cliente no processo de desenvolvimento.

Através da pesquisa pressupõem-se que não existe melhor ou pior entre as metodologias ágeis para gerar produtividade em equipes ágeis, e sim, que fatores devem ser observados em um contexto específico, considerando aspectos como, o tamanho, o tipo do projeto, o prazo de entrega, o escopo, o orçamento disponível, o tamanho da equipe e outras características que determinaram o sucesso e a produtividade considerando práticas organizacionais como o estabelecimento de um terreno comum entre todos os membros da equipe, de regras e objetivos que todos concordam em respeitar, entre outras características, que possam contribuir para que a equipe obtenha maior produtividade, atendendo melhor às necessidades do cliente e do desenvolvedor.

Assim, este trabalho contribui para melhores tomadas de decisões de gerentes de projetos de software, além disso apresenta-se relevante, pois este estudo pode ser utilizado para nortear outras pesquisas teóricas e práticas na perspectiva de métodos ágeis. Em relação a trabalhos futuros, pretende-se realizar um estudo mais aprofundado de quais aspectos devem ser levados em consideração ao escolher um método ágil, e desenvolver e analisar a eficiência de uma ferramenta que possa auxiliar as equipes nessa escolha de forma rápida e prática.

REFERÊNCIAS

- ADELA KUN, O.; IYAMU, T. Translation of Activities in a Global Virtual Teams Software Development. **Journal of Cases on Information Technology**, v. 23, n. 4, p. 1–18, out. 2021. Acesso em: 08 abr. 2022.
- AKBAR, R. Tailoring Agile-Based Software Development Processes. **IEEE Access**, v. 7, p. 139852–139869, 2019. Acesso em: 08 abr. 2022. Disponível em: <https://ieeexplore.ieee.org/document/8849986>. Acesso em: 18 nov. 2021.
- ALQUDAH, M.; RAZALI, R. **An Empirical study of scrumban formation based on the selection of scrum and kanban practices**. International Journal on advanced Science Engineering Information Technology, v.8, n.6, p.15-23, 2018. Disponível em: <https://www.igi-global.com/gateway/article/284568>. Acesso em: 18 jun. 2021.
- AGILE REPORT. **14th Annual State of Agile Report**. 2020. Disponível em: <https://diegocandal.com/wp-content/uploads/2020/10/KM-Brasil-2020-GC-MA-Accepted.pdf>. Acesso em: 18 jun. 2021.
- BASTOS, A. M. .; BASTOS, A. M. **Use of Scrum as a method for optimization in project design. Research, Society and Development**, [S. l.], v. 10, n. 9, p. e40610918329, 2021. DOI: 10.33448/rsd-v10i9.18329. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/18329> . Acesso em: 20 jun. 2022.
- BAXTER, A. L. et al. Collaborative experience between scientific software projects using Agile Scrum development. **Software: Practice and Experience**, v. 52, n. 10, p. 2077–2096. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1002/spe.3120> . Acesso em: 20 out. 2022.
- CALDEIRA, J. E. B. et al. Multiobjective Optimization Using Evolutionary Algorithms in Agile Teams Allocation. **Proceedings of the XVIII Brazilian Symposium on Software Quality**, 28 out. 2019. Acesso em: 02 mar. 2022.
- CERQUEIRA, M. M. B. et al. A Software Development Process for Super Agile Projects. **Proceedings of the XIV Brazilian Symposium on Information Systems - SBSI'18, 2018**. Acesso em: 10 out. 2022.
- CHORAS, M. et al. Measuring and Improving Agile Processes in a Small-Size Software Development Company. **IEEE Access**, v. 8, p. 78452–78466, 2020. Disponível em: <https://ieeexplore.ieee.org/document/9076670>. Acesso em: 20 out. 2022.
- COSTA, A. et al. Team Formation in Software Engineering: A Systematic Mapping Study. **IEEE Access**, v. 8, p. 145687–145712, 2020. Disponível em: <https://ieeexplore.ieee.org/document/9162037>. Acesso em: 20 out. 2022.
- COSTA, A. et al. A Genetic Algorithm-Based Approach to Support Forming Multiple Scrum Project Teams. **IEEE Access**, v. 10, p. 68981–68994, 2022. Disponível em: <https://ieeexplore.ieee.org/document/9807293> . Acesso em: 20 out. 2022.
- DIEBOLD, P.; SCHMITT, A.; THEOBALD, S. Scaling agile. **Proceedings of the 19th International Conference on Agile Software Development: Companion**, 21 maio 2018. Acesso em: 08 abr. 2022
- FABBRI, S. et al. **Improvements in the start tool to better support the systematic review process**, 2016. In: ACM. Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering. Acesso em: 28 out. 2022.

FERNANDEZ-DIEGO, M. et al. An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review. **IEEE Access**, v. 8, p. 166768–166800, 2020. Disponível em: <https://ieeexplore.ieee.org/document/9186107> . Acesso em: 30 out. 2022.

FRONZA, I.; WANG, X. Social loafing prevention in agile software development teams using team expectations agreements. **IET Software**, v. 15, n. 3, 8 abr. 2021. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/sfw2.12019> . Acesso em: 30 out. 2022.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. São Paulo: Atlas, 2007. Acesso em: 30 out. 2022.

KEUP, MEGAN. **O que é Scrumban? Como ele difere do Scrum e Kanban**, 2020. Disponível em : <https://www.projectmanager.com/blog/what-is-scrumban>. Matéria Internet. Acesso em: 26 abr. 2022.

KROPP, M. et al. Satisfaction and its correlates in agile software development. **Journal of Systems and Software**, v. 164, p. 110544, jun. 2020. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0164121220300261?via%3Dihub> . Acesso em: 30 out. 2022.

LUTFIANI, N., Harahap, E. P., Aini, Q., Ahmad, A. D. A. R., & Rahardja, U. Inovasi Manajemen Proyek I-Learning Menggunakan Metode Agile Scrumban. **Jurnal Nasional Informatika Dan Teknologi Jaringan**, 5(1), p. 96–101, 2020. Acesso em: 16, mar. 2022.

MADADIPOUY, K. **An Examination and Evaluation of Agile Methodologies for Systems Development**. Australasian Journal of Computer Science, 2(1), p. 1–17, 2015. Acesso em: 18 mar. 2022.

MKوبا, E.; MARNEWICK, C. Conceptual Framework for Auditing Agile Projects. **IEEE Access**, v. 8, p. 126460–126476, 2020. Disponível em: <https://ieeexplore.ieee.org/document/9136647>. Acesso em: 30 abr. 2022.

MASOOD, Z.; HODA, R.; BLINCOE, K. Real World Scrum A Grounded Theory of Variations in Practice. **IEEE Transactions on Software Engineering**, 2020. Disponível em: <https://ieeexplore.ieee.org/document/9201058> . Acesso em: 20 set. 2022.

MOTA, J. S. Medição da produtividade em times de desenvolvimento de software. **repositório.unb.br**, 9 set. 2021. Disponível em: https://repositorio.unb.br/bitstream/10482/42582/1/2021_JhemesonSilvaMota.pdf . Acesso em: 20 set. 2022.

MOHER, David; STEWART, Lesley; SHEKELLE, Paul. **All in the family: systematic reviews, rapid reviews, scoping reviews, realist reviews, and more**. 2015. Systematic Reviews, vol. 4, no. 168. Acesso em: 25 set. 2022.

MUÑOZ, M.; MEJIA, J.; LAPORTE, C. Y. Implementing ISO/IEC 29110 to reinforce four very small entities of Mexico under an agile approach. **IET Software**, v. 14, n. 2, 1 abr. 2020. Disponível em : <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-sen.2019.0040> . Acesso em: 20 out. 2022.

OLIVEIRA, S. P.; MUNIZ, J. J. **Aplicação do scrum em serviços: análise em uma fabricante de aeronaves**. Revista Produção Online, Florianópolis, SC, v.15, n. 1, p. 276-294, jan./mar. 2015. Disponível em: <https://producaoonline.org.br/rpo/article/download/1777/1256> . Acesso em: 02 de abr. 2022.

OLIVEIRA, E. et al. Surveying the impacts of COVID-19 on the perceived productivity of Brazilian software developers. **Proceedings of the 34th Brazilian Symposium on Software Engineering**, 21 out. 2020. Acesso em: 20 out. 2022.

PIMENTEL, F. S. C.; SALES, J. **Mobilização das habilidades e estratégias metacognitivas por meio dos jogos digitais**. Revista Docência e Ciberultura, [S.L.], v. 5, n. 3, p. 222-242, 2021. Universidade de Estado do Rio de Janeiro. <https://doi.org/10.12957/redoc.2021.61036>. Acesso em: 02 nov. 2022.

PLENGVITTAYA, C.; SANPOTE, D. **Scrumban for teaching at undergraduate program**: A case study from software engineering students, University of Phayao, Thailand. 3rd International Conference on Digital Arts, Media and Technology, ICDAMT 2018, p. 109–114, 2018. Acesso em: 15 mai. 2022.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed. New York: The McGraw-Hill Companies, Inc, 2011. Acesso em: 22 mai. 2022.

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014. Acesso em: 10 abr. 2022.

SANTOS, L. C. M. C.; PRADO, E. P. V.; CHAIM, M. L. Vulnerability detection techniques and tools and their relationship to agile methods and software quality and service models. **Proceedings of the XV Brazilian Symposium on Information Systems**, 2019. Acesso em: 23 jun. 2022.

SEGUNDO, T. F. et al. A Systematic Mapping on CASE Tools Usability Assessment. **Proceedings of the XIV Brazilian Symposium on Information Systems - SBSI'18**, 2018. Acesso em: 23 jul. 2022.

SILVA, J.B; ANASTÁCIO, F.A.M. **Método Kanban como Ferramenta de Controle de Gestão**, v.13. Revista multidisciplinar e psicologia, 2019. Disponível em:<https://doi.org/10.14295/online.v13i43.1575>. Acesso em: 23 jun. 2022.

SILVEIRA, C. B. **Just In Time**: conceito, definição e objetivos. 2018. Disponível em: <https://www.citisystems.com.br/just-in-time-conceito-significado/> . Acesso em: 07 abr. 2022.

SOMMERVILLE, I. 2011 **Engenharia de Software** – Tradução Kalinka Oliveira; Ivan Bosnic. 9º ed. Editora Person Prentice Hall.2011. Acesso em: 12 abr. 2022.

SOTILE, M. **O que é Scrumban?**. Disponível em: <https://blog.pmtech.com.br/scrumban/#> . 2018. Acesso em: 12 abr. 2022

SOUSA, P. R. et al. **Gestão do Fluxo de Pacientes em Internações Relacionadas ao Pronto Socorro: aplicação da Metodologia Kanban**. Revista de Inovação Hospitalar e Inovação em Saúde, Belo Horizonte, v. 14, n. 1, jan./mar. 2017. Acesso em: 08 jun. 2022. Disponível em:<http://dx.doi.org/10.21450/rahis.v14i1.3239>. 2018. Acesso em: 12 abr. 2022.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar. **Metodologias Ágeis: Engenharia de Software sob medida**. São Paulo, 2012. Acesso em: 20 set. 2022.

SCHOEFFEL, P. XP Enigma - Um Jogo Educacional Não Digital para Apoio ao Ensino de Métodos Ágeis: Uma análise temporal da motivação e aprendizagem. **Anais do I Simpósio Brasileiro de Educação em Computação (EDUCOMP 2021)**, 26 abr. 2021. Acesso em: 22 set. 2022.

SCHWABER, K.; SUTHERLAND, J. Guia do Scrum, Um guia definitivo para o Scrum: As regras do jogo. **Scrum Guides**. 2017. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf> f.Acesso em: 13 abr. 2022.

SCHWABER, K.; SUTHERLAND, J. Guia SCRUM. [S. l.]: **Scrum Guides**, 2020.Ebook. 19 p.Scrum-Guide-Portuguese BR-2.0.pdf. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-> . Acesso em: 03 mai. 2022.

SHAMEEM, M. et al. Taxonomy of Success Factors and their Prioritization using Analytic Hierarchy Process for Scaling Agile Process in the Global Software Development. **IET Software**, 20 mar. 2020. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-sen.2019.0196> . Acesso em: 03 out. 2022.

SPIEGLER, S. V.; HEINECKE, C.; WAGNER, S. An empirical study on changing leadership in agile teams. **Empirical Software Engineering**, v. 26, n. 3, 22 mar. 2021. Disponível em: <https://link.springer.com/article/10.1007/s10664-021-09949-5> . Acesso em: 03 out. 2022.

START. **Laboratório de Pesquisa em Engenharia de Software (LaPES)**, Universidade Federal de São Carlos. Disponível em: http://lapes.dc.ufscar.br/tools/start_tool. Acesso em: 18 dez. 2022.

TUBINO, D. F. **Planejamento e Controle da Produção: Teoria e Prática**, 2. ed. São Paulo. Editora Atlas, 2009. 190 p. Acesso em: 19 mai. 2022.

TRIGO, Ramon Alves; BARRETO, Luiz Claudio. **Evolução dos Métodos de Desenvolvimento de Software em Microempresas**. Revista Gestão em Foco - Edição nº 11, 2019. Disponível em: <https://doi.org/10.47820/recima21.v3i10.2121> Acesso em: 06 out. 2022.

VEGA, F. et al. Scrum Watch: a tool for monitoring the performance of Scrum-based work teams. **JUCS - Journal of Universal Computer Science**, v. 28, n. 1, p. 98–117, 28 jan. 2022. Disponível em: <https://lib.jucs.org/article/67593/> . Acesso em: 18 nov. 2022.

WILDT, Daniel et al. **Extreme Programming: práticas para o dia a dia no Desenvolvimento Ágil de Software**, 2015, Ebook.. Acesso em: 20 set. 2022.

WŁODARSKI, R.; PONISZEWSKA-MARAÑDA, A.; FALLERI, J.-R. Impact of software development processes on the outcomes of student computing projects: A tale of two universities. **Information and Software Technology**, v. 144, abr. 2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584921002287?via%3Dihub> .Acesso em: 18 nov. 2022.

YOUNAS, M. et al. Agile Software Development Using Cloud Computing: A Case Study. **IEEE Access**, v. 8, p. 4475–4484, 2020. Disponível em: <https://ieeexplore.ieee.org/document/8943168>. Acesso em: 20 nov. 2022.