

Revista Científica

FACULDADE ATENAS- PARACATU-MG

Ano 2024, V.17, N.1



FACULDADE
ATENAS

www.atenas.edu.br
38 3672-3737

METÓDO ÁGIL SCRUM: Um estudo comparativo entre o método ágil e o método tradicional na prevenção de falha no desenvolvimento de softwares.

Joao Victor Senne Silva Oliveira
Henrique Guedes Pereira
Jean Pablo Monteiro Silva
Anelise Avelar de Araújo
Thais Dias dos Santos

RESUMO

O trabalho apresentado buscou entender e mostrar as diferenças entre os métodos Scrum, que se resume a uma metodologia de geração de valor de forma iterativa e flexível e o Cascata, baseado em um fluxo de trabalho linear e sequencial e pouco flexível, aplicadas no desenvolvimento de software, à fim de identificar qual se torna mais eficiente, além de mostrar também como o processo de Product Discovery e MVP podem auxiliar a identificar pontos críticos e possíveis falhas do produto/projeto antes do seu desenvolvimento. O método cascata possui um grande ponto negativo, que é predefinir os requisitos do software, o que pode trazer sérios problemas futuramente. O trabalho apresenta os riscos e as equipes e, como funciona o Scrum, seus papéis, artefatos e cerimônias. Os dados expostos sobre a eficácia dos dois métodos mostram uma disparidade do método Scrum, tanto em projetos grandes, quanto em projetos pequenos. A junção do processo de Product Discovery e MVP com o método Scrum, tanto na teoria, quanto na prática, entrega a diminuição de falhas que podem ser solucionadas antes mesmo da implementação do produto.

Palavras-chave: Scrum, Cascata, Product Discovery, MVP, eficiência e eficácia.

ABSTRACT

The presented work aimed to understand and demonstrate the differences between the Scrum and Waterfall methods in the context of software development. Scrum is characterized by an iterative and flexible value generation methodology, while Waterfall follows a linear and sequential workflow with limited flexibility. The goal was to identify which method is more efficient. Additionally, the study explored how the Product Discovery process and Minimum Viable Product (MVP) can help identify critical points and potential flaws in the product/project

before development. The Waterfall method has a significant drawback in its predefined software requirements, which can lead to serious problems in the future. The paper discusses the risks associated with both methods and provides insights into how Scrum operates, including its roles, artifacts, and ceremonies. The data presented regarding the effectiveness of the two methods indicates a disparity in favor of Scrum, both in large and small projects. The integration of the Product Discovery process and MVP with the Scrum method, both in theory and in practice, contributes to reducing flaws that can be addressed before the product implementation phase.

Keywords: Scrum, Waterfall, Product Discovery, MVP, efficiency and effectiveness.

1. INTRODUÇÃO

Desde a época em que os softwares começaram a ganhar força, notou-se que o seu desenvolvimento era realizado de forma complexa e desorganizada, sem planejamento, o que muitas das vezes resultava em algo sem qualidade ou que não correspondesse a necessidade dos clientes. Diante disso, Winston W. Royce criou, em meados de 1970, o primeiro modelo de planejamento de desenvolvimento de software, que ficou conhecido como modelo em cascata (SEMEDO, 2012).

O conceito de metodologias ágeis surgiu no início do século XXI, quando dezessete desenvolvedores, como os representantes do método ágil Scrum e eXtreme-Programming, dentre outros, criaram a aliança ágil e assinaram um documento que ficou conhecido como manifesto ágil, uma lista de conceitos e paradigmas a serem seguidos para que a agilidade possa ser aplicada (HIGHSMITH, 2009).

O manifesto ágil pregava que o foco deveria ser a implementação e não a documentação, indivíduos ao invés de processos, colaboração com o cliente mais do que negociação de contratos e a capacidade de mudanças à cima de um plano pré-estabelecido são alguns dos principais conceitos definidos no manifesto. Para Highsmith (2009), a agilidade é a habilidade de equilibrar a flexibilidade com a estabilidade.

Desde então, alguns métodos ágeis vêm ganhando bastante força no ramo de desenvolvimento de software. Um deles é o Scrum. Os criadores do Scrum, ao iniciar a aplicação do método, notaram que equipes menores e multidisciplinares, as consequências eram mais satisfatórias (SEMEDO, 2012). O Scrum conceitua-se em um framework leve, aplicado a

times de desenvolvimento para gerar valor através de soluções adaptativas para problemas complexos (SCHWABER, SUTHERLAND, 2013).

Os projetos desenvolvidos com a metodologia cascata, mostra-nos que, por ser um modelo preditivo, há bastante perda de recursos e prazos ao realizar uma alteração no projeto, seja um processo ou requisito. Mesmo que o produto seja entregue no prazo, nota-se muitas falhas no software, porque os desenvolvedores, muitas das vezes, trabalham com horas extras e sob pressão para entregar no tempo estimado. Com isso, as metodologias ágeis, especificamente o Scrum, veio como resposta ao modelo cascata, pois trata-se de um processo adaptativo, iterativo e incremental (DOS SANTOS SOARES, 2004).

Portanto, este trabalho foca em mostrar como o Scrum pode ser mais eficiente em relação ao modelo cascata, utilizado para inspecionar equipes multidisciplinares e gerar valor em curtos períodos, com um conceito adaptativo, iterativo e incremental, utilizando também algumas técnicas consideradas ágeis que podem mitigar falhas e riscos.

2 MÉTODO SCRUM E O MÉTODO CASCATA

2.1 MÉTODO CASCATA

Para Franco (2007), o avanço tecnológico trouxe consigo a computação, uma forma de processar, armazenar e tratar dados. Desta forma, os hardwares começaram a ter bastante notoriedade, o que gerou bastante investimento no desenvolvimento de sistemas operacionais. No entanto, cada empresa tinha suas necessidades particulares, o que dificultava no desenvolvimento de softwares, esfera esta que cresceu bastante na época, devido à grande procura para o desenvolvimento de softwares. No entanto, esses processos eram feitos de forma desorganizada.

Diante isso, as empresas de desenvolvimento de software passaram a ter bastante foco, e foi por isso que, em meados da década de 70, criou-se o que ficou conhecido como “Método cascata”, desenvolvido por Winston W. Royce, que é um modelo sequencial de desenvolvimento de software, possuindo 6 fases para organizar e planejar o desenvolvimento de software. (MARQUES, 2012).

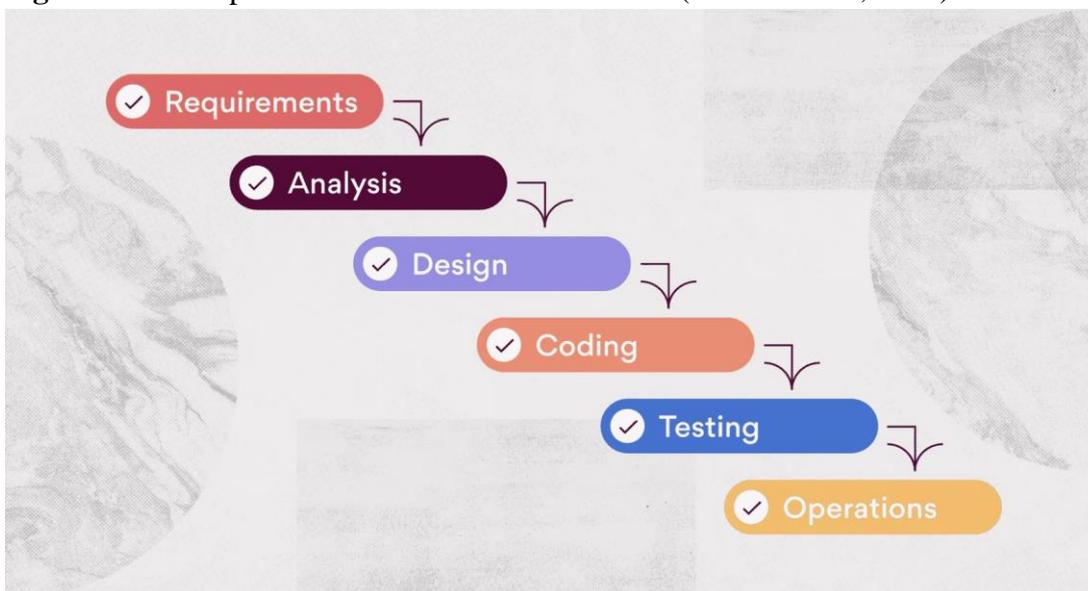
Segundo Semedo (2012), o modelo original do método cascata, proposto por Royce em 1970 era chamado de “Retroalimentação”, uma vez que, a cada etapa concluída, o processo voltava na anterior para interagir ambas as partes.

Para Pressman (2021), o método cascata consiste em analisar os requisitos de cada

cliente, para que o produto desenvolvido satisfaça suas necessidades. Contudo, o que torna o método cascata não tão eficaz, discutido desde sua criação, é que o cliente pode não saber descrever todas as suas necessidades logo no início do projeto, o que pode gerar um certo desconforto deste ao ter o produto entregue.

Outro fator determinante na eficácia do método é o foco, uma vez que, quando aplicado de forma pura, o método cascata busca criar um software bom, ao invés de focar no cliente final, pois este será o usuário de seu produto. Desta forma, o produto pode parecer bom no primeiro instante, mas, o cliente final pode não ficar satisfeito. Na figura 1 a seguir, veremos as etapas da metodologia cascata.

Figura 1 - As etapas do método tradicional Cascata (PRESSMAN, 2021).



Fonte: salesdorado.com, 2023.

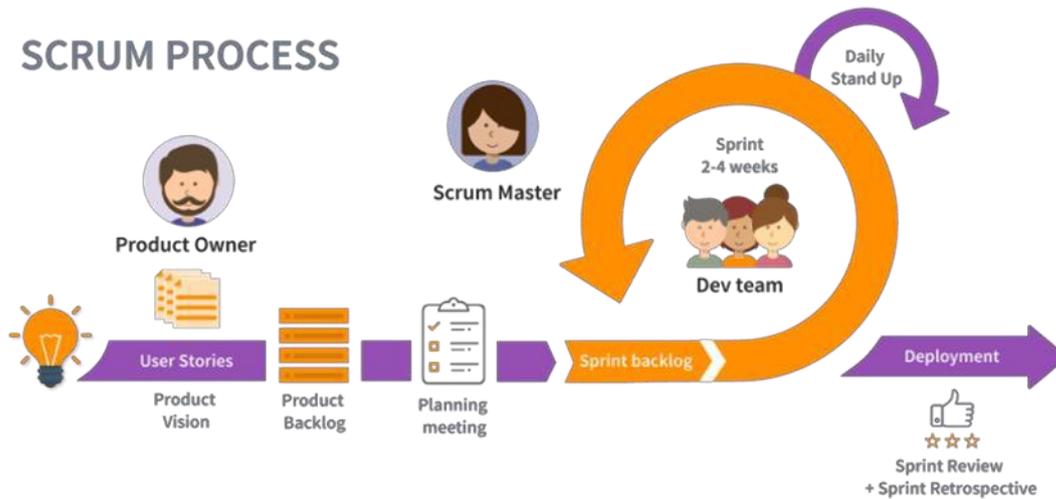
Após o surgimento do método tradicional cascata, as empresas de desenvolvimento e até mesmo os próprios desenvolvedores começaram a duvidar de sua eficácia e começaram a pensar em novas formas de tornar seus processos mais capacitado e seguro.

De acordo com Dos Santos Soares (2004), essa alteração de requisitos em fases elevadas demandava bastante recurso econômico, o que frustrava as partes interessadas, o que consequentemente trouxe o surgimento de um novo modo de planejar e desenvolver softwares, os métodos ágeis. O autor ainda afirma que os métodos ágeis surgiram após uma reunião entre 17 desenvolvedores, que partilhavam da mesma ideia e formaram a Aliança ágil, que consigo, criou o manifesto ágil, um documento composto por práticas e conceitos ágeis, tendo como foco o cliente final, interações, software executável e resposta a mudanças.

Para Sutherland (2013), o Scrum é um framework voltado à times de

desenvolvimento, empresas e qualquer meio que, por meio de um projeto, queira gerar valor a algo complexo. Fundamenta-se de três pilares: Transparência, inspeção e adaptação. Estes pilares provêm do empirismo, expressando a ideia de aprender experimentando, ou seja, através do trabalho em equipe e adaptando a possíveis mudanças durante o desenvolvimento do produto.

Figura 2 - O processo realizado no Scrum.



Fonte: usemobile.com, 2022.

Sutherland (2014) diz também que o Scrum possui alguns papéis, eventos e artefatos, que serão dispostos em tabelas a seguir. A primeira tabela, abordará os papéis presentes no método Scrum:

Tabela 1 - Papéis do Scrum:

NOME	FUNÇÃO
Product Owner	Maximizar o valor do produto, dando clareza sobre o objetivo ao time de desenvolvimento, priorizando e gerenciando o backlog.
Time Scrum	São os responsáveis por criar o que foi priorizado e organizado pelo Product Owner e o Scrum Master, gerando um incremento ao produto a cada Sprint.
Scrum Master	Estabelecer o Scrum. Ele é responsável por manter os conceitos, visão e valores do Scrum no Time Scrum e na organização de todas as partes do time.

Fonte: Scrum.org, 2023.

A segunda tabela, informará os eventos que compõem o Scrum.

Tabela 2 - Eventos do Scrum:

NOME	FUNÇÃO
Planejamento da sprint	É o planejamento feito no início de cada Sprint para mostrar, dar clareza e decidir o que será desenvolvido pelo Time Scrum com as partes interessadas.
Scrum diário	Inspeccionar o trabalho em desenvolvimento do Time Scrum para entender se o Time está caminhando ao objetivo da Sprint.
Revisão da sprint	Inspeccionar o que foi feito durante a Sprint e determinar futuras adaptações. É mostrado o trabalho feito às partes interessadas do produto.
Retrospectiva da sprint	Analisar o que ocorreu de bom e ruim durante a Sprint e propor melhorias para aumentar a qualidade e eficácia do Time Scrum e o produto.
Sprint	É um período de trabalho fixo que engloba todos os eventos do Scrum. É na Sprint que as ideias ganham valor.

Fonte: Scrum.org, 2023.

Por fim, a última tabela mostrará quais artefatos o Scrum possui:

Tabela 3 - Artefatos do Scrum:

NOME	FUNÇÃO
Backlog do produto	É uma lista ordenada das tarefas que, no final, irão compor ou melhorar o produto. O Backlog do Produto pode ser adaptado ao longo do desenvolvimento do produto.
Backlog da sprint	É a lista de itens ordenada, que serão desenvolvidos pelo Time Scrum para gerar o incremento ao produto.
Incremento	Um incremento é quando o time de desenvolvimento entrega um valor ao produto em questão. Cada incremento é relacionado com as entregas (incrementos) anteriores, garantindo que todos funcionem juntos.

Fonte: Scrum.org, 2023.

Para que o Scrum funcione da forma correta, Sutherland (2013) diz que é necessário que todos os personagens presentes no Scrum (papéis), como o Product Owner, Scrum master e time de desenvolvimento, devem pregar os valores do Scrum, que são: Comprometimento, foco, abertura, Respeito e coragem.

Para Sutherland (2013), um time eficaz e eficiente não necessita de diversos indivíduos, a interfuncionalidade pode gerar bons resultados, mas isso não significa que depende de equipes grandes. Ao contrário do que se pensa, a efetividade de uma pequena equipe é de fato melhor do que uma composta por 15 ou mais pessoas. Sutherland prega que a fórmula clássica é de 7 pessoas, ponderando-se para 2 a mais ou 2 a menos.

Além disso, atualmente, outras técnicas ágeis são aplicadas juntas ao Scrum, à fim de melhorar a qualidade do planejamento e identificar e prever riscos e falhas. Dentre estas técnicas, podemos citar o MVP, que se funda no conceito de desenvolver o mínimo de funcionalidades possíveis para que o produto possa funcionar. O MVP exige um trabalho extra e não deve ser entregue a designers e engenheiros para ser validado, é necessário coloca-lo diante dos possíveis clientes para avaliarem o produto (RIES, 2012).

Segundo WERDER.et.al (2016), o vocábulo Product Discovery (Descoberta do Produto) pode ser chamado de fase de geração de ideias. De acordo com Oliveira (2022), a descoberta do produto é fundamental no desenvolvimento de um novo produto, pois ele utiliza técnicas que ajudam a reduzir os riscos de fracasso e aumenta a felicidade do consumidor. Muitos profissionais do ramo da agilidade, onde aplica-se o Scrum, utiliza diversas técnicas para desenvolver a descoberta do produto, como a análise de SWOT (FOFA), mapa de empatia do usuário etc.

Como notamos, o Scrum trabalha de forma iterativa e empírica. Desta forma, para garantir o bom desenrolar do desenvolvimento do software, ele inspeciona o trabalho realizado em diferentes eventos constituídos dentro de sprints. As sprints possuem um time-box, ou seja, ela limita o prazo do desenvolvimento daquele ciclo para ter-se um melhor gerenciamento, que podem ir de 1 semana a 1 mês. Dentro de cada sprint, é possível identificar o progresso em direção ao objetivo da sprint através de várias práticas, como burn-downs, burn-ups ou diagramas (SCHWEBER e SUTHERLAND, 2013).

No início de cada sprint, o Time Scrum cria um plano colaborativo, o que inicia a sprint em questão. Esse start inicial chama-se Planejamento da Sprint, onde é feita por todas as partes interessadas e gerida pelo Product Owner, que irá garantir que todos os indivíduos presentes saibam discutir os itens do Backlog e que tudo que será falado, esteja interligado ao Objetivo do Produto. No Planejamento da Sprint, alguns tópicos são levantados, como: Por que

esta Sprint é valiosa? O que pode ficar pronto nesta Sprint? Como o trabalho selecionado ficará pronto? O Objetivo da Sprint, os itens do Backlog do Produto selecionados para Sprint e o plano para entregá-los são chamados de Backlog da Sprint. (SCHWABER e SUTHERLAND, 2013).

Dado o início da sprint, reuniões diárias são realizadas para inspecionar o progresso em direção ao Objetivo da Sprint e adaptar o Backlog da Sprint quando julgado necessário. Estas reuniões são chamadas de Scrum Diário, realizadas no mesmo horário e local todos os dias durante a sprint. Os Scrum Diários servem para melhorar a comunicação do time, dar clareza ao Product Owner e Scrum Master do que está sendo feito e identificar possíveis impedimentos (SCHWABER e SUTHERLAND, 2013).

Ao final de cada Sprint, o Time Scrum e as partes interessadas se juntam para identificar o que foi entregue de valor e determinar as futuras ações e possíveis mudanças. O Time Scrum quem apresenta o valor gerado na Sprint. O Backlog do Produto também pode ser adaptado caso necessário, para que futuras entregas ocorram (SCHWABER e SUTHERLAND, 2013).

Após a realização de todos os eventos citados anteriormente, o Time Scrum irá realizar o último dos eventos, que tem como propósito identificar pontos de melhoria na qualidade e efetividade, seja em processos, ferramentas, indivíduos e sua Definição de Pronto. Este evento chama-se Retrospectiva da Sprint, onde o Time Scrum discute o que deu certo, o que eles tiveram de problemas/impedimentos e como tudo isso foi ou pode ser solucionado (SCHWABER e SUTHERLAND, 2013).

Para Dos Santos Soares (2004), o que difere as metodologias ágeis do método tradicional são os enfoques e os valores. Ele alega que, o foco nas pessoas e não em processos e os métodos ágeis serem adaptativas, levam-nas a serem mais eficientes. O Scrum, além de pregar o foco no cliente final, não se preocupa tanto em documentação, mas sim na implementação do software.

Outro fator que o torna mais eficiente é o fato de ser adaptativo, ao contrário do tradicional que é preditivo, pois, é mais usual adaptar-se às adversidades encontradas durante o projeto, do que identificar todas as possíveis desvios e impedimentos, o que apresenta alto esforço. Além disso, o ciclo iterativo de entregas palpáveis aos clientes gera maior conforto e satisfação, além de obter feedbacks constantes que possibilitam novas ideias e alterações (DOS SANTOS SOARES, 2004).

3 TÉCNICAS ÁGEIS APLICADAS JUNTO AO SCRUM

3.1 PRODUCT DISCOVERY

O Product Discovery é uma técnica ágil utilizada nas fases iniciais de um projeto, onde sua tradução condiz “Descoberta do Produto”, e serve para entender o problema, descobrir as necessidades dos clientes e criar hipóteses. O Product Discovery é feito por diversas áreas envolvidas naquele projeto, como a equipe de usabilidade, tecnologia e gestão do produto, onde estas áreas visam averiguar e conhecer possíveis concorrentes, oportunidades e problemas de clientes reais. Portanto, a descoberta do produto traz uma matéria prima bem refinada do que irá compor o Product Backlog (PEREIRA, 2023).

Ao buscar-se a eficiência do Product Discovery, nota-se que há um pequeno número de orientação da sua aplicação, porém, os estudos de Trieflinger e seus colaboradores (MUNCH.et.al, 2020) identificaram que o Product Discovery serve para mitigar riscos nas etapas iniciais do desenvolvimento de um software, onde estas etapas trazem diversas dificuldades quanto a identificação real do problema e a falta de clareza do produto para o restante da equipe (PEREIRA, 2023).

3.2 MVP – MINIMUM VIABLE PRODUCT

Para melhorar ainda mais a sua eficiência, o Product Discovery prega a utilização de MVP's (Minimum Viable Product), que consiste em criar uma versão inicial do produto com rápidas iterações e incrementos ao produto, com a finalidade de colher insights para o restante do desenvolvimento do produto. Conseqüentemente, com os insights coletados, a mudança de requisitos, se necessária, pode ser feita, para que não ocorra infelicidade dos usuários na utilização do produto (PEREIRA, 2023).

4 PREVENÇÃO DAS FALHAS E RISCOS ATRAVÉS DO SCRUM

4.1 OS RISCOS E AS EQUIPES

Os principais riscos presentes em projetos de desenvolvimento de software é o quão crítico ele é e como ele pode responder a mudanças. O Scrum, por ser um método ágil é muito bem adaptado a mudanças, pois suas entregas ocorrem em sprints de até 4 semanas e são iterativas. O enfoque ágil traz um ponto onde ele documenta determinados riscos que podem

ocasionar problemas no projeto, fazendo com que tenha um cuidado maior neste ponto, onde o Product Owner irá trabalhar para deixar os requisitos mais refinados e priorizados até chegar nesta etapa de risco do projeto (RIBEIRO, 2020).

Uma pesquisa realizada por PMSurvey.org (2014) diz que cerca de 54,2% das equipes de desenvolvimento de projetos passaram por adversidades quando se tornou necessária a necessidade de mudar o escopo (ALMEIDA, 2020).

Segundo Almeida (2020), a crescente obscuridade do software deve-se a um crescimento significativo na sua utilização para diversos fins, muitas vezes integrados em diferentes áreas, ou a uma mudança na perspectiva do cliente, sendo outro fator que pode contribuir para alterações persistentes na circunferência.

A duração do projeto pode persuadir diretamente em qual metodologia de desenvolvimento será escolhida, pois os recursos serão estimados a partir disso. Diante disso, percebe-se que a implementação do Scrum se torna eficaz em equipes menores, pois equipes maiores exige uma adequação maior ao método, logo, leva mais tempo para preparar os membros destas equipes. Por outro lado, o método cascata escala melhor em equipes e projetos maiores, embora a burocracia também seja maior, levando um risco grande, pois o entrosamento da equipe e a comunicação pode se tornar algo complexo (RIBEIRO, 2020).

De fato, o Scrum aplica-se de forma mais eficiente em equipes menores e interdisciplinares e que, quando há mais que 20 desenvolvedores trabalhando em um mesmo projeto, o método cascata tende a se sobressair. Contudo, atualmente, diversos projetos grandes também dispõem da utilização do Scrum, separando etapas do projeto e migrando equipes pequenas para cada uma destas etapas, tornando assim, equipes mais colaborativas e melhora na comunicação (RIBEIRO, 2020).

Sabemos que as equipes, por possuir indivíduos, possuem características únicas, como habilidade, talentos, cordialidade e comunicação. Além disso, as equipes provêm de autoorganização, autonomia, transparência e interfuncionalidade, fatores estes que o Scrum prega em sua aplicação. Portanto, as equipes ágeis que pregam o Scrum, tendem a ter maior autonomia e organização em relação as grandes equipes presentes na metodologia cascata. Estes fatos nos traz a percepção de que os Scrum Team possuem alta capacidade de mudança, pois os membros podem utilizar qualquer processo para realizar a tarefa, desde que ao final da sprint, ela esteja pronta, fazendo com que lidem de forma autônoma às adversidades que surgem ao longo do projeto (RIBEIRO, 2020).

4.2 CLIENTES COMO FOCO NO DESENVOLVIMENTO

Sabemos que a comunicação é algo bem realizado em times Scrum, logo, a escuta das dores e necessidades dos clientes ou área interessada se torna algo bem eficaz, pois o manifesto ágil prega indivíduos mais do que processos. Com o cliente ou usuário final gerando feedback a todo momento, o Scrum flui bem, desde que o Scrum Team esteja ciente de tudo que esteja acontecendo. Um ponto negativo neste fator é que se o cliente envolvido e o Product Owner não souberem extrair os requisitos da forma correta, pode agregar uma série de consequências, como a falta de transparência com os desenvolvedores e o não entendimento dos requisitos (RIBEIRO, 2020).

4.3 COMPARATIVO ENTRE AS METODOLOGIAS SCRUM X CASCATA

A tabela 4 apresenta um comparativo entre as metodologias Scrum e Cascata, trazendo seus pontos positivos e negativos. Na tabela 4, olharemos a comparação entre estas metodologias.

Tabela 4 - Comparativo entre Scrum x Cascata:

SCRUM	CASCATA
O Scrum tem um ciclo de iteração dividido em sprints, permitindo pequenas entregas de valores.	O processo é dividido em sete etapas distintas, fazendo um desenvolvimento linear.
É uma metodologia flexível a mudanças.	Metodologia estruturada, tornando bastante burocrática e complexa.
Possui diversas entregas que podem ser consideradas projetos diferentes.	O desenvolvimento concluído se torna um único produto.
Há espaço para alteração de requisitos durante o desenvolvimento do projeto.	Não há espaço para mudanças nos requisitos após o seu início.
Por ser um método iterativo, as etapas de design, desenvolvimento, teste e planejamento aparecem diversas vezes.	Cada etapa é concluída apenas uma vez, deixando o processo engessado e por causa disso, pode atrapalhar no prazo.
Foca em indivíduos e coloca o usuário final no centro das atenções.	Foca no software e não é aberto a feedbacks para mudanças.
Eficiente em equipes menores e interdisciplinares, com autonomia e coordenação.	As equipes são grandes e a comunicação não é tão assertiva, logo, sua coordenação também não.

O Product Owner está sempre realizando Product Discovery e validando requisitos durante o desenvolvimento do produto.	A análise de negócios levanta todos os requisitos antes do início do desenvolvimento do software.
---	---

Fonte: <https://www.yodiz.com/blog/agile-vs-waterfall-differences-in-software-development-methodologies/>.

Portanto, nota-se que a aplicação da metodologia Scrum, atualmente, traz mais benefícios e se torna mais eficiente do que o método tradicional Cascata. Também, o Scrum tem um custo-benefício menor, pois sua equipe é pequena e não necessita de gerente de projeto (MOTA, 2018).

Scott Ambler em sua pesquisa realizada em 2008, diz que cerca de 82% das empresas que migraram da metodologia tradicional para a metodologia ágil Scrum, ganharam melhoria na produtividade, em contrapartida, 77% melhoraram seus produtos em termos de qualidade. Outro dado interessante é que 55% dos projetos que empregaram o Scrum, obteve sucesso (ROSSATO, 2018). Na tabela 5 a seguir, avistaremos percentuais em relação ao sucesso destas metodologias.

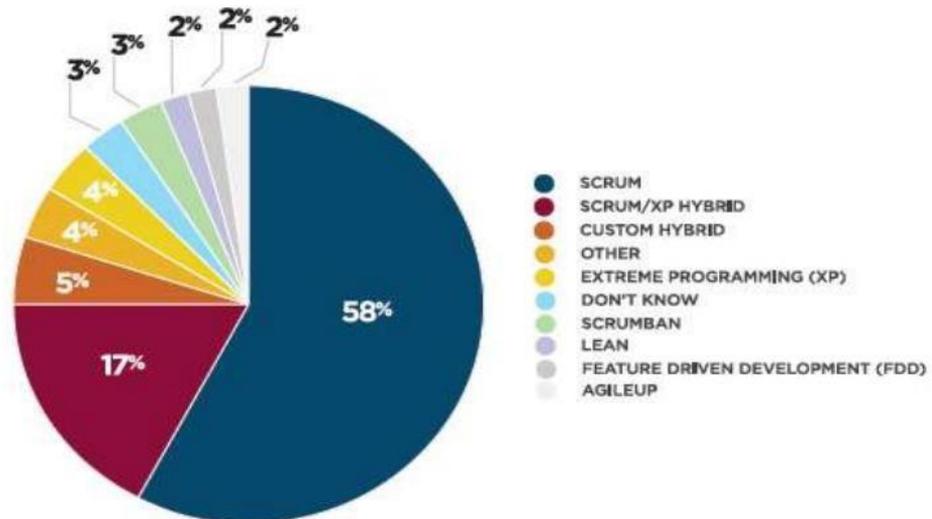
Tabela 5 - Percentual de sucesso das metodologias Scrum e Cascata:

TAMANHO	MÉTODO	BEM-SUCEDIDO	POSTO EM RISCO	CANCELADO
Projeto de todo tamanho	Scrum	39%	52%	9%
	Cascata	11%	60%	29%
Projeto grande	Scrum	18%	59%	23%
	Cascata	3%	55%	42%
Projeto médio	Scrum	27%	62%	11%
	Cascata	7%	68%	25%
Projeto pequeno	Scrum	58%	38%	4%
	Cascata	44%	45%	11%

Fonte: LINCH (2015, p.1).

Constata-se, através da Tabela 3, que o Scrum se sobressai em praticamente todos os tamanhos de projeto, tendo uma baixa somente nos projetos grandes que foram postos em risco, pelo fato do Scrum ser aconselhado a projetos e equipes pequenas. Por conseguinte, o Scrum pode ser aplicado em diversos tamanhos e mesmo assim obterá sucesso maior em relação ao método cascata. Na figura 3 a seguir, contemplaremos as metodologias ágeis mais utilizadas no mundo atualmente.

Figura 3 - Metodologias ágeis mais utilizadas no mundo.



Fonte: Tavares (2010, p.1).

Podemos notar na figura 2, que o Scrum se mostra mais utilizado recentemente em relação as outras metodologias, com um percentual de 58%, seguida do seu uso junto ao XP, atuando de forma hibrida com 17%. A aplicação, também hibrida, do Scrum e Kanban, corresponde a 3%.

5 CONCLUSÃO

Este estudo investigou as características únicas do Scrum, uma metodologia ágil amplamente utilizada, em comparação com abordagens tradicionais, como o modelo em cascata. A literatura revisada e os dados apresentados mostram que o Scrum é caracterizado pela flexibilidade, adaptabilidade e uma abordagem de desenvolvimento iterativo que permite uma resposta mais ágil às mudanças nos requisitos do projeto. Uma comparação do Scrum com métodos tradicionais, baseada em teorias consagradas como o Manifesto Ágil, trabalhos de autores como Schwaber e Cockburn e pesquisas empíricas de Dybå e Dingsøyr, mostra as principais vantagens do Scrum na prevenção de erros. A flexibilidade do Scrum para se adaptar às novas necessidades e a ênfase na interação contínua com o cliente permitem que os problemas sejam identificados precocemente, reduzindo assim os custos associados a atrasos na correção de bugs. A introdução do Product Discovery dentro do Scrum, conforme proposto por Cagan, enfatiza a importância de entender as necessidades do cliente desde os estágios iniciais de um projeto, e fornece informações importantes mesmo antes do desenvolvimento ter efetivamente começado. Portanto, este estudo destaca a eficácia do Scrum na prevenção de

erros no desenvolvimento de software e mostra que combinar metodologias ágeis como o Scrum com práticas como a descoberta de produtos pode reduzir riscos e melhorar a qualidade dos produtos de software. Isso sugere que é um método promissor a ênfase nas práticas ágeis, embora reconheça as contribuições das metodologias tradicionais, demonstra uma adaptabilidade significativa para enfrentar os desafios dinâmicos dos ambientes modernos de desenvolvimento de software.

REFERÊNCIAS

ALMEIDA, Renan Rosado de. **Comparativo entre o gerenciamento tradicional e o ágil de projetos de desenvolvimento de software**. Gerência de Projetos de Tecnologia da Informação Unisul Virtual, 2020.

Cagan, M. (2017). **Inspired: How to Create Products Customers Love**. Wiley.

DOS SANTOS SOARES, Michel. **Comparação entre metodologias Ágeis e tradicionais para o desenvolvimento de software**. INFOCOMP Journal of Computer Science, v. 3, n. 2, p. 8-13, 2004.

DOS SANTOS SOARES, Michel. **Metodologias ágeis extreme programming e scrum para o desenvolvimento de software**. Revista Eletrônica de Sistemas de Informação, v. 3, n. 1, 2004.

FRANCO, Eduardo Ferreira. **Um modelo de gerenciamento de projetos baseado nas metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta**. 2007. Tese de Doutorado. Universidade de São Paulo.

HIGHSMITH, Jim. **Agile project management: creating innovative products**. Pearson education, 2009.

LINCH, Jennifer, 2015 - Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch. Disponível em <<https://www.infoq.com/articles/standish-chaos-2015>>. Acesso em 13 de setembro de 2017.

MACHADO, Marcos; MEDINA, Sérgio Gustavo. **SCRUM–Método Ágil: uma mudança cultural na Gestão de Projetos de Desenvolvimento de Software**. Revista Científica Intraciência, Faculdade do Guarujá–UNIEESP, v. 1, n. 1, p. 58-71, 2009.

MARQUES, André. **Metodologias ágeis de desenvolvimento: Processos e Comparações**. São Paulo, 2012.

MOTA, Edmarson Bacelar. **GANHOS DE PRODUTIVIDADE EM PROJETOS DE SOFTWARE UTILIZANDO METODOLOGIA ÁGIL FRAMEWORK SCRUM VS MODELO TRADICIONAL PMBOK**. 2018. Tese de Doutorado. Fundação Getúlio Vargas.

NAGAI, Ronaldo Akiyoshi; SBRAGIA, Roberto. **As origens da metodologia ágil: de onde saímos e onde estamos? Uma revisão sistemática da literatura**. Revista de Gestão e Projetos, v. 14, n. 1, p. 11-41, 2023.

OLIVEIRA, Francisco Alan Jefferson. **O efeito da utilização do Product Discovery para encontrar novas oportunidades e criar produtos em empresas de tecnologia**. 2022.

PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. **Entendendo Scrum para gerenciar projetos de forma ágil**. Mundo PM, v. 1, p. 3-11, 2007.

PEREIRA, Chris Herbert Berenguer. **A computação afetiva como recurso para estudos de usabilidade em práticas de product discovery: uma abordagem utilizando Machine Learning**. 2023.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software-9**. McGraw Hill Brasil, 2021.

RIES, Eric. **A startup enxuta**. Leya, 2012.

RIBEIRO, MATHEUS BARON. **CENÁRIOS DE APLICAÇÃO DAS METODOLOGIAS TRADICIONAIS E ÁGEIS NO DESENVOLVIMENTO DE SOFTWARE**.

ROSSATO, Cleder Adriano. Comparativo entre o método ágil e o tradicional. Governança de Tecnologia da Informação-Unisul Virtual, 2018.

Schwaber, K., & Beedle, M. (2002). **Agile Software Development with Scrum**. Prentice Hall.

SCHWABER, Ken; SUTHERLAND, Jeff. **La guía de Scrum. Scrumguides**. Org, v. 1, p. 21, 2013.

SEMEDO, Maria João Moreno et al. **Ganhos de produtividade e de sucesso de Metodologias Ágeis VS Metodologias em Cascata no desenvolvimento de projectos de software**. 2012.

SILVA, Edson Coutinho; LOVATO, Leandro Alvarez. **Framework Scrum: eficiência em projetos de software**. Revista de Gestão e Projetos, v. 7, n. 2, p. 01-15, 2016.

Disponível em: <https://www.scrum.org/learning-series/what-is-scrum>. Acesso e, 06 de abril de 2023.

SUTHERLAND, Jeff. **SCRUM: A arte de fazer o dobro de trabalho na metade do tempo**. Leya, 2014.

TAVARES, Giselle, 2010 - **Metodologias ágeis em uma organização baseada em linha de produto**. Disponível em <<http://www.devmedia.com.br/uso-de-metodologiasageis-em-umaorganizacao-baseada-em-linha-de-produto-artigo-revista-engenharia-desoftware-magazine38/21662>>. Acesso em 26 de outubro de 2017.

WERDER, Karl; ZOBEL, Benedikt; MAEDCHE, Alexander. **PDISC—Towards a Method for Software Product DIScovery: Type: Exploratory Paper**. In: Software Business: 7th International Conference, ICSOB 2016, Ljubljana, Slovenia, June 13-14, 2016, Proceedings 7. Springer International Publishing, 2016. p. 47-62.