



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIA E TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA METALÚRGICA E DE MATERIAIS**

**RAYANE MENDES ROCHA**

**ANÁLISE DA APLICAÇÃO DA METODOLOGIA SCRUM PARA  
GERENCIAMENTO DE PROJETOS DE SOFTWARE DE FORMA REMOTA EM  
UMA INSTITUIÇÃO DE CIÊNCIA E TECNOLOGIA DE FORTALEZA DURANTE A  
PANDEMIA DO COVID-19**

**FORTALEZA**

**2022**

RAYANE MENDES ROCHA

ANÁLISE DA APLICAÇÃO DA METODOLOGIA SCRUM PARA GERENCIAMENTO DE PROJETOS DE SOFTWARE DE FORMA REMOTA EM UMA INSTITUIÇÃO DE CIÊNCIA E TECNOLOGIA DE FORTALEZA DURANTE A PANDEMIA DO COVID-19

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Metalúrgica e de Materiais da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia Metalúrgica.

Orientador: Prof. Dr. Marcelo José Gomes da Silva.

FORTALEZA/CE

2022

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

R576a Rocha, Rayane Mendes.

Análise da aplicação da metodologia SCRUM para gerenciamento de projetos de software de forma remota em uma instituição de ciência e tecnologia de Fortaleza, durante a Pandemia do COVID-19 / Rayane Mendes Rocha. – 2022.

53 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Metalúrgica, Fortaleza, 2022.

Orientação: Prof. Dr. Marcelo José Gomes da Silva.

1. Desenvolvimento de software. 2. Metodologia Ágil. 3. Gerenciamento de Projetos. 4. SCRUM. I. Título.

CDD 669

---

RAYANE MENDES ROCHA

ANÁLISE DA APLICAÇÃO DA METODOLOGIA SCRUM PARA GERENCIAMENTO DE PROJETOS DE SOFTWARE DE FORMA REMOTA EM UMA INSTITUIÇÃO DE CIÊNCIA E TECNOLOGIA DE FORTALEZA DURANTE A PANDEMIA DO COVID-19

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Metalúrgica e de Materiais da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia Metalúrgica.

Aprovada em: xx/xx/xxxx.

BANCA EXAMINADORA

---

Prof. Dr. Marcelo José Gomes da Silva (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Jorge Luiz Cardoso  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Candido Jorge de Sousa Lobo  
Universidade Federal do Ceará (UFC)

Dedico este trabalho a minha mamãe,  
Regina Célia e meu vovô João Batista.

## **AGRADECIMENTOS**

A minha família por todo o apoio durante toda a minha graduação.

A minha companheira de vida, por todo cuidado e força compartilhada.

Aos meus queridos amigos de curso que foram de extrema importância e suporte durante esta jornada. André, Brenna, Bruna, Eric, João, Marcellus, Maykel e Veimar.

Aos meus companheiros de Centro acadêmico, Empresa Júnior e Centro de Empreendedorismo que contribuíram para minha formação para além da sala de aula.

Aos amigos feitos durante meu estágio, que sem dúvida impactaram positivamente minha carreira profissional e área de atuação.

Aos professores participantes da banca examinadora pelo tempo e sugestões.

“Inclua no seu método de trabalho a possibilidade de mudança, descoberta e inovação.” (SUTHERLAND, 2016, p.190).

## RESUMO

As metodologias ágeis surgiram como uma alternativa para a até então forma tradicional de gerenciar projetos de software. Esse tipo de metodologia considera mais importante pessoas e entregas de valor do que documentações e processos. Destas metodologias, a mais difundida atualmente é a metodologia Scrum, que vem sendo aplicada não só para gerenciamento de projetos de software como vários outros tipos de projetos. Com o início do período pandêmico que parou o mundo por conta do COVID-19, a adoção do trabalho remoto se tornou a opção mais segura para contornar a situação de contágio da doença no dia a dia do trabalho e muitas empresas tiveram de se adaptar a esta realidade. Um dos setores que mais aderiram e mantêm a pretensão de continuar aderindo o teletrabalho pós pandemia é o setor de tecnologia da informação, que trabalha majoritariamente com desenvolvimento de software e conseqüentemente em muitos casos aderindo o Scrum como forma de gerenciar seu desenvolvimento. Diante disso, este trabalho tem como objetivo a análise da aplicação desta metodologia em um ambiente controlado e identificar suas características, dificuldades e possíveis pontos de melhoria para o estudo de caso investigado. Todos os insumos foram coletados e tratados a partir de entrevistas com membros de um time do instituto eleito para análise.

**Palavras-chave:** Desenvolvimento de software, SCRUM, Metodologia Ágil, Gerenciamento de Projetos.

## **ABSTRACT**

Agile methodologies emerged as an alternative to the traditional way of managing software projects. This type of methodology considers people and value deliverables more important than documentation and processes. About these methodologies, the most widespread today is the Scrum methodology, which has been applied not only to software project management but also to other types of projects. With the beginning of the pandemic period that stopped the world because of COVID 19, the adoption of remote work became the safest option to circumvent the situation of contagion in the day to day of work and many companies had to adapt to this situation. One of the sectors that most adhered and maintains the intention to continue adhering to remote work after the pandemic is the information technology sector, which works mostly with software development and consequently in many cases adhering to Scrum as a way to manage its development. Therefore, this work aims to analyze the application of this methodology in a controlled environment and identify its characteristics, difficulties and possible points of improvement for the investigated case study. All inputs were collected and treated from interviews with members of an institute team elected for the analysis.

**Keywords:** Software development, SCRUM, Agile Methodology, Project Management.

## LISTA DE FIGURAS

Figura 1 – Relação entre requisitos, prazos e custos de um projeto .....	21
Figura 2 – Fases do modelo Cascata .....	22
Figura 3 – Metodologia tradicional x metodologia ágil .....	25
Figura 4 – O Processo Scrum .....	26
Figura 5 – Ciclo de vida da sprint .....	29

## LISTA DE TABELAS

Tabela 1 – Caracterização de funcionários entrevistados .....	34
Tabela 2 – Modelo teórico adotado para questionário .....	35
Tabela 3 – Visão sobre cada papel do time .....	38
Tabela 4 – Ferramentas de auxílio para reuniões remotas .....	46

## LISTA DE ABREVIATURAS E SIGLAS

PO	<i>Product Owner</i>
SM	<i>Scrum Master</i>
Dev	Desenvolvedor
TI	Tecnologia da Informação

## LISTA DE SÍMBOLOS

% Porcentagem

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>14</b>
<b>1.1</b>	<b>Contextualização</b> .....	<b>14</b>
<b>1.2</b>	<b>Objetivos</b> .....	<b>16</b>
<b>1.2.1</b>	<b><i>Objetivo Geral</i></b> .....	<b>16</b>
<b>1.2.2</b>	<b><i>Objetivos específicos</i></b> .....	<b>16</b>
<b>1.3</b>	<b>Justificativa da pesquisa</b> .....	<b>16</b>
<b>1.4</b>	<b>Definição do problema</b> .....	<b>17</b>
<b>1.5</b>	<b>Estrutura do trabalho</b> .....	<b>18</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>19</b>
<b>2.1</b>	<b>Desenvolvimento de software</b> .....	<b>19</b>
<b>2.2</b>	<b>Metodologia tradicional no desenvolvimento de software</b> .....	<b>21</b>
<b>2.3</b>	<b>Metodologias ágeis no desenvolvimento de software</b> .....	<b>23</b>
<b>2.3.1</b>	<b><i>O modelo Scrum aplicado no desenvolvimento de software</i></b> .....	<b>25</b>
<b>2.3.1.1</b>	<b><i>Time Scrum</i></b> .....	<b>27</b>
<b>2.3.1.2</b>	<b><i>Eventos e artefatos Scrum</i></b> .....	<b>28</b>
<b>2.4</b>	<b>O trabalho remoto e seus aspectos na atualidade</b> .....	<b>30</b>
<b>3</b>	<b>ESTUDO DE CASO</b> .....	<b>32</b>
<b>3.1</b>	<b>Metodologia</b> .....	<b>32</b>
<b>3.2</b>	<b>Definição da unidade de análise definição de problema</b> .....	<b>32</b>
<b>3.3</b>	<b>Condução do estudo de caso único</b> .....	<b>34</b>
<b>3.4</b>	<b>Análise em relação ao time Scrum</b> .....	<b>36</b>
<b>3.5</b>	<b>Análise em relação aos pilares Scrum</b> .....	<b>39</b>
<b>3.5.1</b>	<b><i>Backlog do produto</i></b> .....	<b>40</b>
<b>3.5.2</b>	<b><i>Eventos</i></b> .....	<b>41</b>
<b>3.6</b>	<b>Análise em relação aos valores Scrum</b> .....	<b>46</b>
<b>4</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>48</b>
	<b>REFERÊNCIAS</b> .....	<b>49</b>
	<b>ANEXO A – MATRIZ DE STORY POINTS</b> .....	<b>53</b>
	<b>APÊNDICE A – PERGUNTAS UTILIZADAS NAS ENTREVISTAS</b> .....	<b>54</b>

# 1 INTRODUÇÃO

Este capítulo apresenta os objetivos gerais da pesquisa, ou seja, uma análise da aplicação da metodologia Scrum para o gerenciamento de projetos de software, sendo eles executados de forma remota. O estudo foi realizado em uma instituição de ciência e tecnologia do estado do Ceará. Além disso, pretende mostrar os objetivos específicos da pesquisa, bem como a sua justificativa e estruturação.

## 1.1 Contextualização

Segundo Schwab (2016), diante do cenário atual de desenvolvimento, pode-se perceber que existem mudanças profundas em alguns importantes setores da tecnologia, que, por sua vez, são marcadas por mudanças nos modelos de negócio, sobretudo, pela descontinuidade dos operadores e pela reformulação da produção, do consumo, dos transportes e dos sistemas logísticos. Ademais, observa-se uma mudança de paradigma no modo como trabalhamos e nos comunicamos, bem como na maneira de nos expressarmos, nos informarmos e nos divertirmos (SCHWAB, 2016). Como consequência desta veloz transformação tecnológica, observou-se o aparecimento de algumas estruturas que visavam solucionar problemas específicos, entre elas o uso das metodologias ágeis (FADEL; SILVEIRA, 2010).

De acordo com Beck et al. (2010), em 2001, durante um encontro entre 17 líderes que trabalhavam no contra fluxo dos padrões da indústria de software, foi discutido formas de trabalho, de maneira a chegar a uma nova metodologia de produção de software a qual pudesse ser usada por todos eles e em outras empresas, substituindo os modelos tradicionais de desenvolvimento. O intuito da utilização de metodologias ágeis, independente de qual seja, é o foco no usuário e o retorno rápido das percepções de uso desse software pelos mesmos, de forma a focar principalmente na entrega funcional ao invés da entrega na documentação do processo, fator comum em metodologias tradicionais (FADEL; SILVEIRA, 2010).

Nesse contexto, surge o Scrum como uma das metodologias para gerenciamento de projetos de fabricação de automóveis e de produtos de consumo que, posteriormente, passou a ser adotada para desenvolvimento de software.

Segundo Fadel e Silveira (2010), o Scrum não define uma técnica específica para o desenvolvimento de software durante a etapa de implementação, uma vez que se concentra em descrever como os membros da equipe devem trabalhar para produzir um sistema flexível, num ambiente de mudanças constantes. A ideia central do Scrum é que o desenvolvimento de sistemas envolve diversas variáveis (ambientais e técnicas) e elas possuem grande probabilidade de mudar durante a execução do projeto (por exemplo: requisitos, prazos, recursos, tecnologias etc.) (FADEL; SILVEIRA, 2010).

Com o avanço tecnológico, novas formas de flexibilização das relações de trabalho têm se disseminado, entre as quais o teletrabalho (BARROS; SILVA, 2010). Durante a pandemia acarretada pelo coronavírus que afetou a população global em inúmeros aspectos, entre eles a relação com o trabalho, a adaptação a esse panorama desafiador surge como uma medida urgente em parcela considerável de empresas e de instituições, a fim de se manterem ativas e potencialmente concorrentes no mercado (BRIDI et al., 2020). De fato, um dos principais ramos adeptos dessa movimentação, principalmente por possuir processos majoritariamente digitais, encontram-se na área da tecnologia da informação, a qual, entre outras atribuições, atua no desenvolvimento de softwares. No Brasil, uma pesquisa realizada pela Betania Tanure Associados (BTA), que contou com a participação de 359 companhias, aponta que trabalhar de casa (*home office*) virou a realidade de 43% das empresas brasileiras durante a pandemia do COVID-19, e que a maior dificuldade encontrada foi a adaptação das atividades presenciais para as virtuais (>60%), dados que justificam a situação adversa enfrentada pelo país (BIGARELLI, 2020).

Quando surgiu nos anos 80, a metodologia Scrum foi desenvolvida em um contexto no qual o modelo de trabalho adotado era de forma presencial e, apesar de ter sofrido evoluções ao longo dos anos, com a finalidade de se adaptar diante das necessidades encontradas, durante a pandemia da COVID-19 houve uma necessidade específica de migração de muitos times de desenvolvimento de software para a modalidade *home office*. Assim, esta pesquisa tende a investigar a aplicação da metodologia Scrum para gerenciamento de projetos de software de forma remota em uma instituição de ciência e tecnologia de fortaleza.

## **1.2 Objetivos**

### **1.2.1 Objetivo Geral**

Este trabalho objetiva analisar a aplicação da metodologia Scrum para gerenciamento de projetos de software de forma remota em uma instituição de Ciência e Tecnologia.

### **1.2.2 Objetivos Específicos**

- a) Descrever o gerenciamento de projetos de software dando enfoque para a metodologia Scrum.
- b) Avaliar a adoção da metodologia Scrum em formato remoto para o gerenciamento de projetos de software.
- c) Identificar as dificuldades de utilização da metodologia Scrum para o gerenciamento de projetos de software de forma remota.

## **1.3 Justificativa da pesquisa**

Segundo Soares (2004), as metodologias ágeis para desenvolvimento de software são uma resposta às chamadas metodologias pesadas ou tradicionais, que passaram a ter visibilidade a partir dos anos 2000. Muito embora as metodologias ágeis apresentem variações em termos de práticas e ênfases, tais recursos compartilham algumas características, como desenvolvimento iterativo e incremental, comunicação e redução de produtos intermediários, como documentação extensiva, existindo, por sua vez, maiores possibilidades de atender aos requisitos do cliente, que muitas vezes são mutáveis (SOARES, 2004). Em contrapartida, apesar da popularização destes métodos ágeis para gerenciamento de projetos, não existem muitos documentos que façam análises das consequências das aplicações destes *frameworks*.

De acordo com a Digital.ai (2020), uma das principais empresas mundiais de consultoria em transformação digital, em seu relatório anual de status de agilidade, é destacado que 94% das empresas analisadas praticam algum tipo de metodologia ágil. Dessa parcela, 86% são relacionados ao setor de desenvolvimento

de software e >75% se utilizam do Scrum ou da combinação dele com características de outras metodologias. Destarte, devido ao grande número de indivíduos no setor da tecnologia que hoje utilizam a metodologia Scrum, inúmeras demandas de estudo naturalmente vão surgindo com o objetivo de analisar a utilização dessa forma de gerenciamento de projetos.

Ademais, não é só a utilização do Scrum que vem crescendo nos últimos anos. Uma crescente adaptação de empresas para adoção do trabalho para a modalidade remota decorrente da pandemia do COVID 19 vê-se como algo que já faz parte do planejamento das empresas mesmo após o período pandêmico, em que, de acordo com a World Economic Forum (2020), as principais ações que as empresas pretendem tomar como parte de seus planejamentos estratégicos são: Acelerar a digitalização (92%) - atividade diretamente ligada ao desenvolvimento de software - e fornecer maiores oportunidades de trabalho remoto aos seus colaboradores (88%).

Dessa forma, diante da escassez de literaturas que explicam sobre o impacto da utilização de metodologias ágeis como alternativa para gerenciamento de projetos e sobre o modo que se espera que o mercado brasileiro se comporte no cenário pós pandemia para a adoção do trabalho presencial, esta pesquisa visa trazer mais informações acerca deste assunto, sobretudo no que diz respeito à metodologia Scrum e sua aplicação de forma remota dentro de um estudo de caso único.

#### **1.4 Definição do problema**

Em uma empresa de tecnologia focada em pesquisa e desenvolvimento de software localizada em Fortaleza, no estado do Ceará, denominada neste estudo de Instituto Ômega, surgiu a necessidade da migração de todos os setores para o trabalho remoto, essencialmente por conta da pandemia durante o início de 2020. A companhia trabalha de maneira projetizada para outras empresas, o que significa que, por meio de contratos fechados, o instituto Ômega desenvolve softwares para as contratantes externas. Tal processo de gerenciamento destes projetos, ocorre por intermédio da metodologia Scrum. Ademais, é conhecido que, para o gerenciamento de projetos mediante o uso desta metodologia, são necessárias diversas reuniões

características do ciclo de entrega do software, as quais são denominadas cerimônias. Dessa forma, a partir de um dado momento, todo o gerenciamento do projeto, bem como as cerimônias que tangem a metodologia e os demais repasses entre a equipe passaram a ser de forma virtual.

## **1.5 Estrutura do trabalho**

O trabalho foi estruturado em 4 capítulos compreendendo seções de introdução, referencial teórico, estudo de caso e conclusão, o qual cada um possui seus respectivos detalhamentos.

O primeiro capítulo conta com introdução sobre o tema da pesquisa, motivação e objetivos específicos e gerais a serem alcançados neste trabalho.

No segundo capítulo são apresentados os conceitos teóricos sobre gerenciamento de projetos de software e a metodologia Scrum para esta finalidade.

O terceiro capítulo mostra os resultados das análises sobre a utilização, dificuldades e benefícios da metodologia Scrum quando aplicada de forma remota em um instituto de tecnologia de Fortaleza. As informações foram coletadas via entrevistas com os funcionários desta empresa. Este capítulo também contém a metodologia e o material utilizado para a coleta e tratamento de dados.

O quarto e último capítulo conta com a conclusão, considerações finais acerca da pesquisa realizada e recomendações para trabalhos futuros.

Neste trabalho ainda contém referências bibliográficas a fim de consultas.

## 2. REFERENCIAL TEÓRICO

Este capítulo aborda os tópicos referentes ao referencial teórico. A seção apresenta conceitos relevantes que foram utilizados para o desenvolvimento do trabalho. O item 2.1 discorre sobre o conceito de desenvolvimento de software, o item 2.2 descreve sobre o gerenciamento tradicional de projetos de software e o 2.3 sobre esse gerenciamento de forma ágil, dando enfoque para a utilização da metodologia Scrum. O último item trata sobre o trabalho remoto durante a pandemia do COVID 19 e suas perspectivas.

### 2.1 Desenvolvimento de software

Com o avanço a passos largos da tecnologia da informação e sua influência na sociedade, é comum que cada vez mais seja expansível o mercado ligado a ela, fator que pode ser percebido observando a transformação digital que empresas e governos vêm sofrendo nos últimos anos aplicando soluções tecnológicas para problemas tradicionais, a exemplo do agendamento de algum serviço. Parte desse avanço é pautado no desenvolvimento de softwares para a entrega de produtos.

Segundo Sommerville (2011):

Engenharia de software é importante por dois motivos:

1. Cada vez mais, indivíduos e sociedades dependem dos sistemas de software avançados. Temos de ser capazes de produzir sistemas confiáveis econômica e rapidamente.
2. Geralmente é mais barato, a longo prazo, usar métodos e técnicas da engenharia de software para sistemas de software, em vez de simplesmente escrever os programas como se fossem algum projeto pessoal. Para a maioria dos sistemas, a maior parte do custo é mudar o software depois que ele começa a ser usado.

Para Ribeiro et al. (2017), um software é todo programa rodado em um computador, celular ou dispositivo que permita ao mesmo tempo executar suas funções, que, por sua vez, diferenciam-se da produção manufatureira principalmente pelo alto dinamismo que envolve o ambiente e o processo de produção de software. Nos ambientes fabris, geralmente tem-se um processo linear, estruturado e rígido e na maioria dos casos toda a produção é feita por máquinas que são preparadas e

programadas para realizar uma única tarefa, atendendo assim um estágio específico do processo (RIBEIRO et al., 2017).

Segundo Filho (2003), o desenvolvimento de software se preocupa com o software enquanto produto. Assim como os produtos industriais, os softwares têm um ciclo de vida: ele é concebido a partir da percepção de uma realidade; desenvolvido, transformando-se em um conjunto de itens entregue a um cliente; entra em operação, sendo usado dentro de algum processo de negócio e sujeito a atividades de manutenção, quando necessário; e finalmente, é retirado de operação, ao final de sua vida útil. A principal diferença entre os produtos de manufatura e os softwares se mostra no que tange às alterações feitas no produto, uma vez que os softwares podem sofrer mudanças sempre que necessário.

Desse modo, para atender ao ciclo de vida do desenvolvimento de um sistema, independente da forma em que o mesmo for implementado, deve atender a quatro atividades fundamentais: Especificação de software; Projeto e implementação de software; Validação de software e Evolução de software (SOMMERVILLE, 2011).

Normalmente, o desenvolvimento de software é feito dentro de um projeto, que tem uma data de início, uma data de fim, uma equipe e outros recursos. (FILHO, 2003). A entrega de um produto de software com qualidade, dentro do prazo e custos esperados no projeto é visto como um grande desafio para as organizações que trabalham com esse viés tecnológico (BARCELLOS, Monalessa Perini, 2004).

Destarte, requisitos, prazos e custos formam os vértices de um triângulo crítico (Figura 1). Aumentos de requisitos levam a aumentos de prazos ou de custos, ou de ambos. Reduções de requisitos podem levar, algumas vezes, a reduções de prazos ou de custos (FILHO, 2003). Assim, determinar qual metodologia será utilizada no gerenciamento e entrega desse software pode ditar o sucesso do mesmo.

Figura 1 - Relação entre requisitos, prazos e custos de um projeto



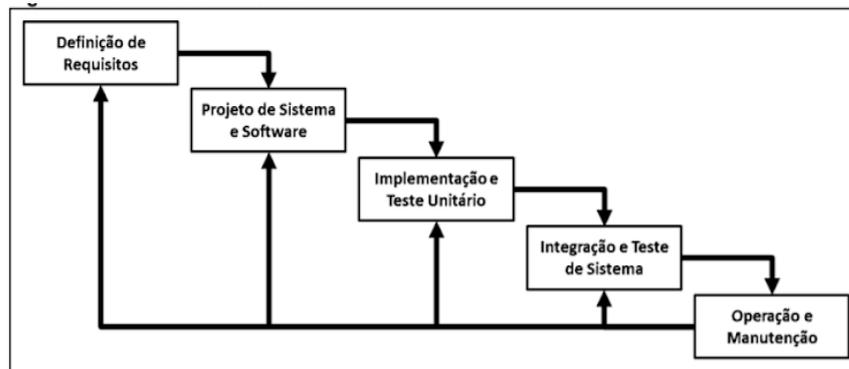
Fonte: Filho (2003)

## 2.2 Metodologia tradicional no desenvolvimento de software

Um método de software, frequentemente chamado de metodologia, foca em como navegar em cada fase do ciclo de vida do projeto e como representar as fases dos produtos (ALMEIDA, 2017). Metodologias tradicionais são definidas como modelos de processo prescritivos, ou seja, aqueles que se baseiam em uma descrição de como as atividades são feitas (WAZLAWICK, 2013).

Conhecido como Ciclo de Vida Clássico, o Modelo Cascata foi o primeiro modelo de desenvolvimento de software a ser publicado baseado nos processos da engenharia de sistemas e seu nome foi baseado na forma sequencial de suas fases (ROYCE, 1970). Este modelo tem como ponto de partida o planejamento prévio de cada etapa, no qual é redigido um escopo do andamento de cada tarefa antes do seu início (Figura 2). O modelo Cascata é constituído de 5 fases, representadas na Figura 1, as quais são executadas de maneira sequencial e sistemática, ou seja, uma após a outra, sendo que uma nova fase só é iniciada após a conclusão de sua antecessora e, por este motivo, é essencial que exista um comprometimento inicial do processo e que o cliente seja específico em seus requisitos, uma vez que só ao final do projeto será analisada uma mudança no processo (STANKIEWICZ, 2017).

Figura 2 - Fases do modelo Cascata



Fonte: Sommerville (2011)

Durante a primeira fase, a definição de requisitos, são elucidadas todas as funcionalidades, serviços e restrições do sistema. Nessa fase, todos os responsáveis pelo projeto se reúnem com os clientes e é realizado o levantamento de requisitos, como por exemplo quais pessoas usarão o software, como ele será utilizado, e todas as demais informações que sejam relevantes para a construção do software (STANKIEWICZ, 2017).

Já na segunda etapa, no Projeto de Sistema e Software, é definida uma arquitetura geral do sistema, todos os requisitos são agrupados e estudados para a organização do modelo de software que será gerado. Além disso, nesta fase são definidos os passos que o sistema irá seguir, como a definição de escopo, distribuição de tarefas, definição de cronograma e verificação das etapas para o desenvolvimento (SCHNELL, 2018).

Na terceira fase, de Implementação e Teste Unitário, ocorre a codificação propriamente dita, que consiste em colocar em um modo de representação através de símbolos por meio de alguma linguagem de programação, além da escrita do manual de utilização do sistema para o cliente (SCHNELL, 2018).

Na quarta fase, de Integração e Teste do Sistema, todas as unidades individuais que foram desenvolvidas são integradas e testadas como um sistema completo para garantir que tenham sido atendidos todos os requisitos determinados pelo cliente na fase de definição de requisitos e, em seguida, é entregue ao cliente (SCHNELL, 2018).

Por fim, na quinta e última etapa, Operação e Manutenção, o sistema é instalado e colocado em uso. Nessa etapa é comum que apareçam erros e omissões que não foram possíveis de serem visualizadas anteriormente, quando os requisitos originais foram colhidos. Dessa forma, são feitas as alterações necessárias para que os erros sejam corrigidos, o que pode implicar na repetição de estágios anteriores do processo, já que o Modelo Cascata define que os requisitos que precisam sofrer alterações só poderão ser atualizados ao término de todas as fases do modelo (STANKIEWICZ, 2017).

Ao longo dos anos surgiram derivações do modelo Cascata, como o Modelo Sashimi, Modelo V, Modelo W, entre outros (WAZLAWICK, 2013), porém pelo fato de ser uma metodologia baseada em documentação, atividades bem descritas e dependentes entre si, nesse modelo descrito cada etapa necessita da aprovação de um ou mais documentos, geralmente assinado pelas partes interessadas e uma etapa só deve ser iniciada se a fase anterior tiver sido concluída. Na prática, esses estágios não são sempre dependentes entre si e se sobrepõem e alimentam uns aos outros de informações. Ademais, problemas são encontrados durante todo o projeto, seja com os requisitos ou seja durante a codificação. O processo de software não é um modelo linear simples, mas envolve o feedback de uma fase para outra (SOMMERVILLE, 2011).

### **2.3 Metodologias ágeis no desenvolvimento de software**

Nesse contexto, antes do surgimento das metodologias ágeis, havia uma visão generalizada de que a melhor maneira para conseguir o melhor software era por meio de um planejamento cuidadoso do projeto, qualidade da segurança formalizada, do uso de métodos de análise e do processo de desenvolvimento de software rigoroso e controlado (SOMMERVILLE, 2011). Toda a burocracia, dependência entre etapas e grandes chances de retrabalho, por conta do feedback tardio dos usuários finais até que todas as etapas se cumpram, gerou uma insatisfação na comunidade de engenharia de software com essas abordagens pesadas no desenvolvimento de sistemas. Tal insatisfação com essas abordagens pesadas da engenharia de software impulsionou um grande número de desenvolvedores de software a proporem, na década de 1990, novos 'métodos ágeis' (SOMMERVILLE, 2011).

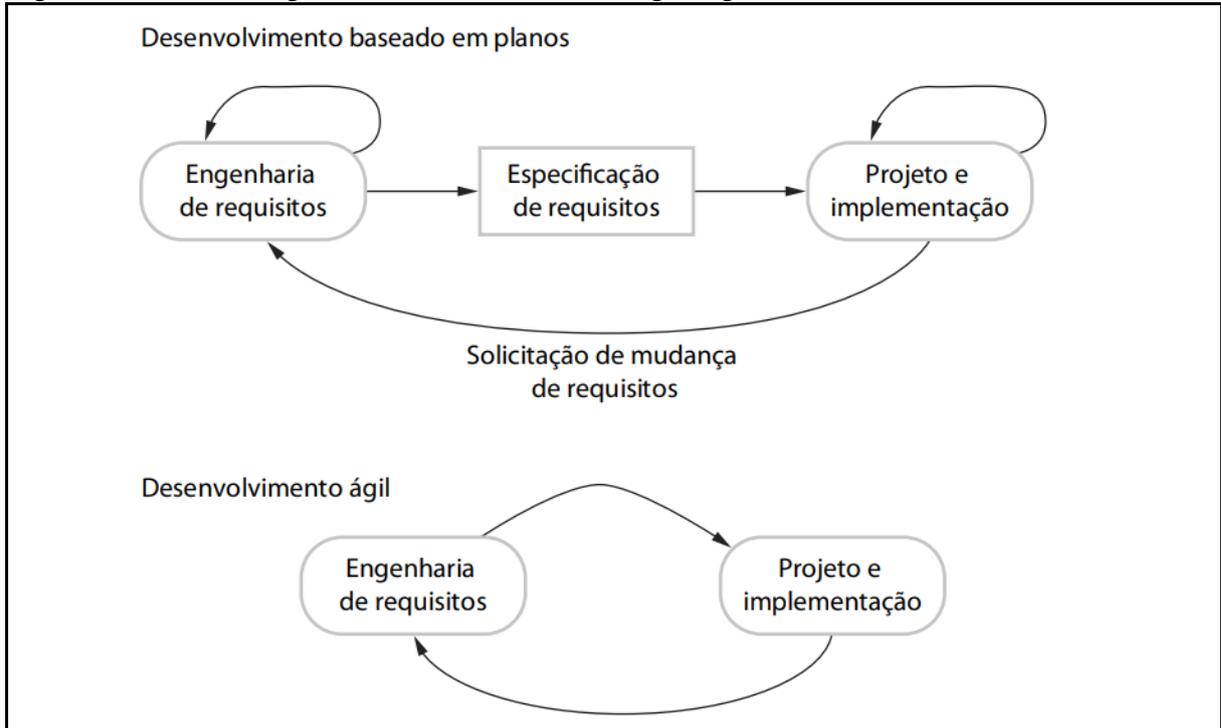
Todavia, apesar de os métodos ágeis serem usualmente mais leves, é errado entendê-los como modelos de processos menos complexos ou simplistas. (WAZLAWICK, 2013). O termo “Metodologias Ágeis” tornou-se popular quando dezessete especialistas em processos de desenvolvimento de software estabeleceram princípios comuns compartilhados por todos esses métodos. Dessa forma, foi, então criada a Aliança Ágil e o estabelecimento do “Manifesto Ágil” (Beck et al. (2010). Os conceitos chave do “Manifesto Ágil” são:

- a) Indivíduos e interações ao invés de processos e ferramentas.
- b) Software executável ao invés de documentação.
- c) Colaboração do cliente ao invés de negociação de contratos.
- d) Respostas rápidas a mudanças ao invés de seguir planos.

Vale ressaltar que, segundo Soares (2004), o Manifesto Ágil não ignora processos e ferramentas, documentação e negociação de contratos, ou seja, essas características, que já existiam no Modelo Cascata, seguem presentes, contudo, o que as diferencia das metodologias tradicionais são o enfoque e os valores. A ideia das metodologias ágeis é o enfoque nas pessoas e não em processos ou algoritmos.

Dessa maneira, um paralelo entre metodologias ágeis e metodologias tradicionais como resumo do que já foi descrito nos dois últimos tópicos pode ser visto na figura abaixo (Figura 3):

Figura 3 - Metodologia tradicional x metodologia ágil



Fonte: Sommerville (2011)

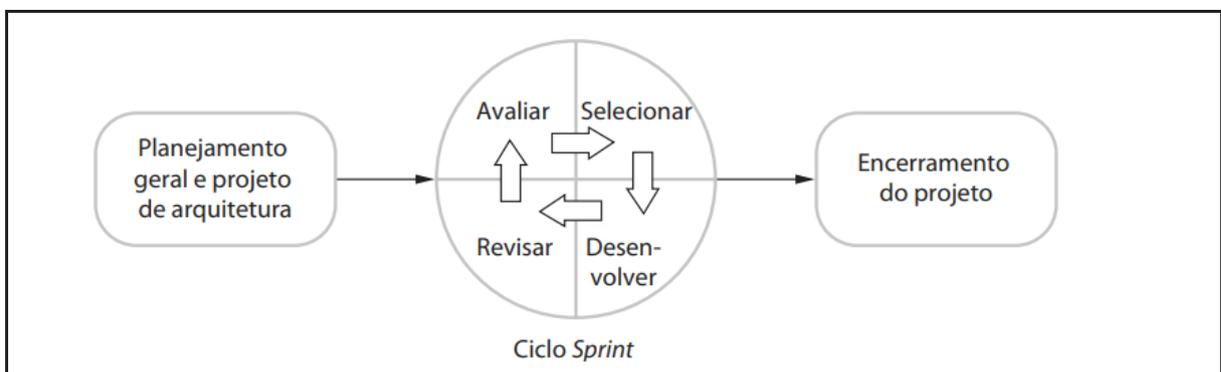
### 2.3.1 O modelo Scrum aplicado no desenvolvimento de software

O Scrum é um dos *frameworks* do contexto ágil utilizado na construção de projetos de softwares, usado em trabalhos complexos nos quais não é possível prever tudo o que irá ocorrer e oferece um conjunto de práticas que torna tudo visível. Isso permite aos praticantes do Scrum saber exatamente o que está acontecendo ao longo do projeto e fazer os devidos ajustes para manter o projeto se movendo ao longo do tempo visando alcançar os seus objetivos (PEREIRA et al., 2007). Diferente do Modelo Cascata, no Scrum não é necessário esperar até a última etapa do processo para realizar as mudanças, tendo em vista que o Scrum é um *framework* vivo que permite constantes alterações no escopo do projeto, caso surja a necessidade para tal, dessa forma, evita-se que as falhas e erros sejam levados até o final do desenvolvimento, possibilitando que seja alterado o mais cedo possível.

Segundo Carvalho et al. (2020), o Scrum utiliza as abordagens iterativa e incremental, de forma a possibilitar que a equipe de desenvolvimento faça modificações e busque melhorar o produto durante sua criação. Contém seis princípios para auxiliar no desenvolvimento do projeto, princípios esses que são

fundamentais para a correta aplicação do seu framework: controle de processos empíricos, auto-organização, colaboração, priorização baseada em valor, time-box e desenvolvimento. O Scrum é iterativo, em observância aos valores de comprometimento da equipe, coragem, foco, transparência e respeito (Carvalho et al., 2020). A figura abaixo (Figura 4) mostra o diagrama de forma simplificada o processo Scrum de gerenciamento, o qual após um entendimento geral do que será o produto e definições iniciais de arquitetura e requisitos do software, se dá início a sequência cíclica de ações com pequenas entregas e de forma incremental ao produto final até o encerramento do projeto.

Figura 4 - O Processo Scrum



Fonte: Sommerville (2011)

A primeira aparição do termo Scrum é datada de 1986 em um artigo de Hirotaka Takeuchi e Ikujiro Nonaka publicado na Harvard Business Review. Nesse artigo os autores fazem um comparativo da forma de desenvolvimento de software com esportes e citam que até então as metodologias aplicadas eram parecidas com uma corrida de revezamento, na qual para dar início a uma “corrida” o desenvolvedor anterior deveria ter terminado a “corrida” anterior, quando na verdade o desenvolvimento de software assemelhava mais a um jogo de rúgbi, o ideal era que toda a equipe jogasse ao mesmo tempo, passando as tarefas para frente ou para trás até que chegasse no objetivo (SUTHERLAND, 2016). Scrum é também o nome de uma jogada famosa no rúgbi.

Ademais, a primeira versão do Scrum surgiu em 1993 na Easel Corporation por Ken Schwaber em conjunto com Jeff Sutherland. A iniciativa de adaptação do gerenciamento de software a partir do artigo “The New New Product Development Game” (O novo jogo para o desenvolvimento de novos produtos,

tradução livre) de Takeuchi e Nonaka foi criada na com o objetivo de melhorar a entrega do produto em que estavam trabalhando. Em 1995, após a metodologia ter sido validada nos corredores da Easel, Jeff Sutherland e Ken Schwaber publicaram o artigo “SCRUM Development Process” (Processo de desenvolvimento SCRUM, tradução livre), o qual apresentava o processo adotado por eles (SUTHERLAND, 2016).

Decerto, a partir de 2010 um documento público chamado “The Scrum Guide” (O Guia Scrum, tradução livre) é mantido por Schwaber e Sutherland em uma plataforma web com o intuito de apresentar as adaptações e “regras do jogo” para a utilização da metodologia Scrum. O documento é atualizado periodicamente, tendo sua última atualização ao documento feita em 2020.

#### 2.3.1.1 *Time Scrum*

Para entender como funciona o modelo Scrum, é necessário elaborar um panorama dos papéis desempenhados por cada integrante do time, que consiste em um *Product Owner*, um *Scrum Master* e uma equipe de desenvolvedores (Schwaber; Sutherland, 2020). Dentro do time Scrum não há subgrupos ou hierarquias, é uma unidade coesa de profissionais focados em um objetivo de cada vez, a meta do produto. Schwaber e Sutherland (2020) definem o *Product Owner* como o dono do produto, ele é o responsável por aumentar o valor do produto a ser desenvolvido e também do trabalho realizado pelo time de desenvolvimento. O modo como o *Product Owner* atua pode variar de empresa para empresa e, independente da forma, o papel do PO também é responsável pelo gerenciamento eficaz do *Product Backlog* (Backlog do produto, tradução livre). Em seguida, encontra-se o *Scrum Master*, sendo este o responsável por estabelecer o Scrum conforme definido no Guia do Scrum. Eles fazem isso ajudando todos a entender a teoria e a prática do Scrum, tanto no time Scrum quanto na organização, e são responsáveis também por eliminar os impedimentos que possam aparecer ao longo da execução do projeto (Schwaber; Sutherland, 2020). Por fim, o time Scrum conta também com uma equipe de desenvolvedores, o qual se encontra responsável por fazer nascer o software por meio de sua codificação, cabendo no seu leque 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).

### 2.3.1.2 *Eventos e artefatos Scrum*

Nesse sentido, após o planejamento geral do projeto, inicia-se, de fato o ciclo de desenvolvimento do produto até a sua entrega. Sendo assim, durante as atividades cíclicas de avaliar, selecionar, desenvolver e revisar (Sommerville, 2011), existem eventos que regem esses ciclos, onde existe: a *sprint*, as reuniões diárias (*daily scrum*), a revisão da *sprint* (*sprint review*), a retrospectiva da *sprint* (*sprint retrospective*) e o planejamento da *sprint* (*sprint planning*). Segundo Schwaber e Sutherland (2020), *sprints* são eventos de duração fixa de um mês ou menos e ao final desse prazo é entregue parte do produto “pronto”, de maneira que uma versão incremental do produto que pode ser utilizada. Durante as *sprints*, nenhuma mudança é feita que coloque em risco a meta da *sprint*, de forma que a qualidade do que está sendo desenvolvido não diminui e o escopo pode ser esclarecido e renegociado com o *Product Owner* conforme mais é aprendido sobre o projeto. Por conseguinte, as reuniões diárias, segundo Schwaber e Sutherland (2020), são eventos diários de 15 minutos para os desenvolvedores do time, cujo objetivo é de verificar o que foi realizado no dia anterior, o que será feito no atual e quais as dificuldades que possivelmente foram encontradas. Para reduzir a complexidade, é realizado no mesmo horário e local, todos os dias úteis da *sprint*.

O planejamento da *sprint* é uma reunião que ocorre antes do início da *sprint* e os envolvidos no processo se reúnem para discutir os principais pontos a serem desenvolvidos para alcançar o produto desejado pelo cliente. A revisão da *sprint* tem como propósito inspecionar o resultado da alcançado naquele ciclo e determinar as adaptações futuras, o time apresenta os resultados de seu trabalho para as principais partes interessadas e o progresso em direção a meta do produto é discutido, de forma a analisar se o software está de acordo com o planejado de início e se de fato atende às expectativas (Schwaber; Sutherland, 2020). Já a retrospectiva da *sprint* consiste em planejar maneiras de aumentar a qualidade e a eficácia da *sprint*, tendo que para isso, o time relembre como foi a realização da última *sprint* com relação a indivíduos, interações, processos e ferramentas

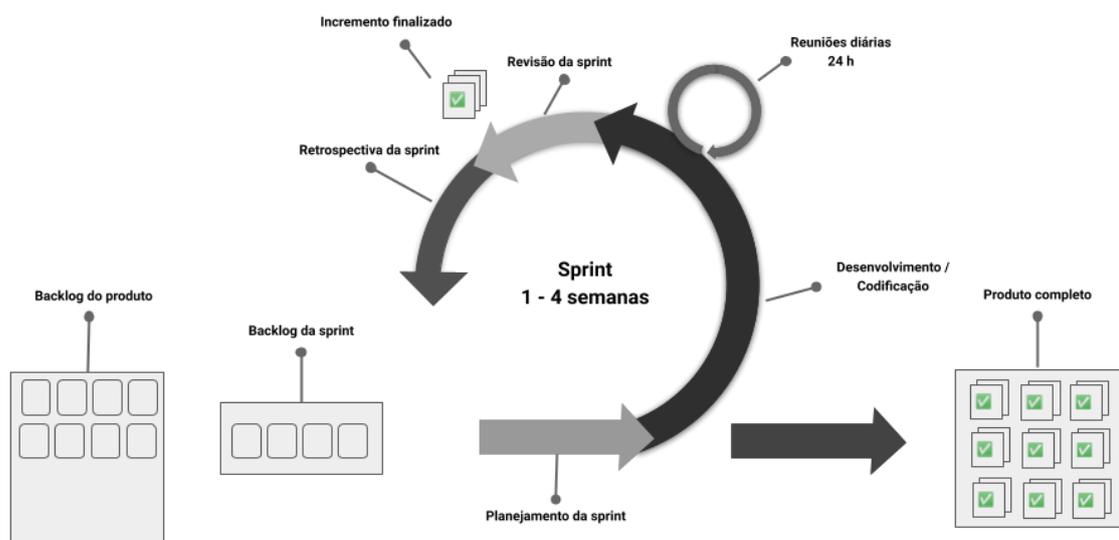
utilizadas. Nela, se discute o que deu certo durante a, quais os problemas encontrados e como esses problemas foram ou não resolvidos.

No que diz respeito ao *Scrum*, também é importante explicar sobre seus artefatos, estes são: O backlog do produto, que consiste em uma lista ordenada e emergente do que é necessário para a entrega do produto final; O backlog da *sprint*, que consiste em um conjunto de itens do backlog do produto para serem trabalhados dentro de uma única *Sprint*, e por último, mas não menos importante, o incremento propriamente dito do produto ao final de uma *sprint* (Schwaber; Sutherland, 2020).

O backlog do produto apresenta requisitos de alto nível, bastante voltados para a meta a ser alcançada para a entrega final e às necessidades diretas do cliente. Já o *sprint backlog* apresenta uma visão menor desses requisitos presente no backlog do produto, de forma mais refinada e mais esclarecida, facilitando a maneira como a equipe vai desenvolvê-los e realizar o incremento ao final da *sprint* (WAZLAWICK, 2013).

Abaixo é apresentado de forma visual (Figura 5) a dinâmica da *sprint* com seus artefatos e eventos:

Figura 5 - Ciclo de vida da *sprint*



## 2.4 O trabalho remoto e seus aspectos na atualidade

O trabalho remoto, ou teletrabalho, surgiu em um contexto em que se preconizava uma nova lógica de interação humana, na medida em que o avanço da globalização e o acelerado desenvolvimento da tecnologia da informação e da comunicação rompiam as fronteiras espaço-temporais estabelecidas pela modernidade (BARROS; SILVA, 2010). Desse modo, os cenários de empregabilidade, que antes estavam estritamente relacionados com a presença física do funcionário na empresa vêm mudando, sobretudo naquelas empresas cujo trabalho pode ser executado de maneira remota sem nenhum prejuízo, com o uso de ferramentas virtuais para o auxílio da comunicação.

Outrossim, a tecnologia como um todo permite que o trabalhador tenha acesso ao trabalho de qualquer lugar do mundo, não dependendo de sua posição física, fator que reverbera na forma como o trabalhador atual exerce sua função programada, não necessariamente estando em um só local para poder executar tais funções de trabalho. Nesse sentido, uma das características laborais principais de pessoas que trabalham com desenvolvimento de software, é trabalhar apenas via computador, uma vez que seu trabalho depende de elaborar códigos em diferentes linguagens de programação, e, para isso é necessário, em suma, um computador. À vista disso, a adoção do teletrabalho para pessoas que trabalham com desenvolvimento de software faz-se bastante lógica e tem se mostrado cada vez mais comum, como apresenta a CIO (2016), que, em um estudo pré-pandêmico, mostra um crescimento de 24% no número de funcionários que aderiram ao trabalho remoto em uma pesquisa.

Além disso, durante o contexto da pandemia do COVID-19, diversas empresas foram obrigadas a mudar seu estilo de trabalho para proteger seus funcionários e garantir o isolamento social. Se antes da crise da COVID-19 o trabalho remoto/home-office vinha despontando como uma tendência, desde 2020, com a recomendação do isolamento e distanciamento social, a modalidade ganhou um espaço ainda maior nos diversos segmentos e setores econômicos do país e do mundo. Muitos trabalhadores tiveram, dentro do trabalho que realizam, que se adequar a uma nova forma de executar suas atividades laborais (BRIDI et al., 2020). Nesse sentido, a utilização de trabalho humano será a área em que a pandemia

causará seus maiores impactos, pois se trata de uma crise que afeta praticamente todos os segmentos da força de trabalho, bem como o local de trabalho diretamente (CHAHAD, 2021). Assim, pode-se observar uma tendência a reorganização do panorama de trabalho, trazendo muito mais flexibilidade e adaptação por parte das empresas, principalmente no meio da tecnologia. De fato, pode-se dizer que a pandemia acelerou 3 tendências que poderão persistir após seu término, com diferentes implicações para o trabalho, sendo elas: A aceleração do trabalho remoto híbrido em casa/escritório; O equilíbrio entre o *e-commerce* (comércio eletrônico) e o *delivery economy* - o sentimento generalizado em que os clientes esperam entrega de mercadorias de baixo custo, rápida e altamente transparente –; Tendência de empresas para a acelerar a automação e as inteligências artificiais para enfrentar a COVID-19, elevando a utilização permanente da robotização (CHAHAD, 2021).

Com isso, percebe-se que estudar os aspectos do modelo Scrum, uma das metodologias de gerenciamento de projetos mais adotadas, no contexto do remoto é um ponto a ser bastante trabalhado, uma vez que a adoção do home office já vinha se esboçando antes da pandemia e encontrou nela um poderoso acelerador para sua consolidação.

De acordo com Fischer (2002), o desempenho que se espera das pessoas no trabalho e o modelo de gestão correspondente são determinados por fatores internos e externos ao contexto organizacional. Isso quer dizer que as consequências do trabalho gerado nas organizações são resultadas das escolhas de gestão feitas pela empresa: tecnologia adotada, estratégia de organização do trabalho, cultura e estrutura organizacional são alguns exemplos. Nesse âmbito, alguns aspectos abordados nesse estudo, tais como a adoção do modelo remoto de trabalho e a escolha do Scrum como ferramenta facilitadora da condução dos projetos, dizem respeito a essas escolhas feitas pela gestão do instituto Ômega.

### **3. ESTUDO DE CASO**

Este capítulo apresentará a metodologia utilizada na pesquisa e a condução do estudo de caso, para então apresentar as análises realizadas a partir de informações e tratamento de dados qualitativos obtidos através de entrevistas. O estudo de caso visa analisar a aplicação da metodologia Scrum a partir dos pilares, valores e definições de time Scrum a partir do referencial teórico.

#### **3.1 Metodologia**

Segundo Godoy (1995), um fenômeno pode ser melhor compreendido no contexto em que ocorre e do qual é parte, devendo ser analisado numa perspectiva integrada. Para tanto, o pesquisador vai a campo buscando captar o fenômeno em estudo a partir da perspectiva das pessoas nele envolvidas, considerando todos os pontos de vista relevantes. Por terem essas características, esse trabalho de conclusão de curso pode ser avaliado como abordagem qualitativa, sendo ela de caráter exploratório, uma vez que visa proporcionar maior familiaridade com o problema da pesquisa por meio da elaboração de questões de pesquisa ou desenvolvimento de hipóteses explicativas para os fatos e fenômenos a serem estudados (GIL, 2003; LAKATOS; MARCONI, 2003). Além disso, essa pesquisa foi conduzida como um estudo de caso único que se trata de um estudo em profundidade de uma unidade de interesse, a fim de trazer uma análise dos dados colhidos no referido estudo de caso.

#### **3.2 Definição da unidade de análise e definição de problema**

Segundo Siglenton (1988), unidades de análise, são os objetos ou eventos os quais as pesquisas sociais se referem, o que ou quem será descrito, analisado ou comparado. Decerto, a definição da unidade de análise e, portanto, do caso analisado, encontra-se relacionada à maneira como as questões iniciais da pesquisa foram definidas (Yin, 2001). Sendo assim, o presente trabalho tem como objetivo geral analisar a aplicação da metodologia Scrum para gerenciamento de projetos de software de forma remota, sobretudo, durante a pandemia do COVID-19, em uma instituição de ciência e tecnologia em Fortaleza. Dessa forma, a unidade de análise foi um projeto pertencente ao instituto Ômega apresentado anteriormente.

O instituto Ômega fica localizado na capital do Ceará: Fortaleza e está no mercado desde 2011. A empresa é uma Instituição de Ciência e Tecnologia (ICT), estruturada na forma de uma associação civil, voltada para a promoção da inovação através da pesquisa e desenvolvimento. Tal instituto apresenta mais de 600 funcionários que hoje colaboram para a melhor organização desta instituição e funciona de forma projetizada, em outras palavras, empresas terceiras contatam o instituto Ômega e contratam uma equipe por um determinado período de tempo, a fim de receber um produto ao final do projeto, de tal modo que as equipes são formadas de acordo com a necessidade de elaboração e execução do projeto.

Ademais, uma das equipes está desenvolvendo um produto para um dos principais clientes do instituto Ômega, o qual vamos chamar de empresa X. Esse produto é um sistema de monitoramento de erros em aplicações pertencentes ao ecossistema tecnológico da própria empresa.

Esse projeto teve início no começo de 2020, onde a configuração de modelo de trabalho era a seguinte: Instituto Ômega situado em Fortaleza, onde toda a sua equipe de desenvolvimento atuava de forma presencial e a empresa X situada no sul do Brasil, e por ser uma multinacional, o alinhamento de alguns requisitos se davam com membros da empresa em várias partes do mundo. Há alguns anos o instituto Ômega adota a metodologia Scrum como forma de gerenciamento de seus projetos de software, onde até então alguns momentos marcantes do projeto como planejamento inicial e entregas do produto ocorriam de forma presencial, com a equipe contratada viajando até a sede da contratante e realizando esses encontros, tratando as demais necessidades em reuniões online. Com a ocorrência da pandemia do COVID19, um novo cenário de trabalho surgiu e, a partir de então, os encontros que antes ocorriam de forma presencial com o cliente, tiveram de ser adaptados para a modalidade online, assim como toda a rotina do time Scrum que agora estariam trabalhando de forma remota.

Além da adaptação do time Scrum para trabalhar de casa, houve a necessidade de algumas trocas na equipe e contratação de novos membros por conta da rotatividade dos empregados, dessa forma, alguns indivíduos do time deste referido projeto nunca chegaram a se conhecer pessoalmente ou trabalhar junto de forma presencial.

Nesse cenário descrito acima, este trabalho buscou analisar como a metodologia Scrum foi aplicada no trabalho remoto, bem como os benefícios e as

dificuldades encontradas pela equipe, visto que, até o presente momento, a maior parte da literatura foca no Scrum aplicado de forma presencial.

A equipe do projeto analisada nesta pesquisa é composta por 7 pessoas onde são descritas como: *Product Owner*, *Scrum Master*, *Dev 1*, *Dev 2*, *Dev 3* e *Dev 4*. As informações gerais do time Scrum do projeto analisado se encontram abaixo na Tabela 1:

Tabela 1 - Caracterização de funcionários entrevistados

Membro	Idade	Formação acadêmica	Tempo de instituto	Você já trabalhou com SCRUM antes deste projeto? Por quanto tempo?	Tempo de atuação no papel exercido hoje dentro do time
PO	26	Ensino médio	1 ano e 2 meses	Não	1 ano e 2 meses
SM	36	Análise e Desenvolvimento de Sistemas	10 anos e 2 meses	Sim, há 8 anos	2 anos
Dev 1	32	Engenharia da computação	10 meses	Sim, há 6 anos	12 anos
Dev 2	30	Engenharia de controle e automação e Análise e desenvolvimento de sistemas	1 ano e 3 meses	Sim, mas adaptado com outras metodologias ágeis	2 anos e 7 meses
Dev 3	25	Ciência da computação	4 meses	Não	1 ano
Dev 4	25	Ciência da computação	1 ano e 4 meses	Sim, 3 anos	3 anos

Fonte: Produzido pelo autor

Tendo em vista a ambientação descrita acima e analisando as características do time estudado, é possível destacar que 83% dos membros possuem formação superior na área de tecnologia da informação e mais de 50% já trabalhou com Scrum ou adaptações dele durante suas carreiras. A vista disso, é relevante pontuar que, com exceção do *Scrum Master*, todos os demais membros do projeto adentraram no instituto Ômega após o início da pandemia de COVID-19, conseqüentemente já em andamento do modelo remoto para o gerenciamento deste projeto.

### 3.3 Condução do estudo de caso único

Sendo assim, para a coleta de dados deste estudo de caso, a fonte de evidências utilizada para a pesquisa e para a análise foi obtida por meio de entrevistas realizadas com os membros do time Scrum analisado no instituto Ômega.

Desse modo, como forma de elucidação do objetivo desta pesquisa, foi desenvolvido pelo autor um roteiro com questões (Ver Apêndice A), a partir do referencial teórico sintetizadas na Tabela 2 a seguir:

Tabela 2 - Modelo teórico adotado para questionário

<b>Fatores base do Scrum</b>	<b>Variáveis de pesquisa</b>	<b>Referências</b>
Pilares do Scrum	Artefatos Scrum (A1) Eventos Scrum (A2) Adaptação do trabalho (A3)	Ken Schwaber e Jeff Sutherland (2020)
Valores do Scrum	Compromisso (B1) Respeito (B2) Coragem (B3) Foco (B4) Abertura (B5)	Ken Schwaber e Jeff Sutherland (2020)
Time Scrum	Composição do time Scrum (C1) Papéis e responsabilidades do time Scrum (C2)	Ken Schwaber e Jeff Sutherland (2020)

Fonte: Produzido pelo autor

De fato, para melhor entendimento, verifica-se necessário a explicação sobre como o questionário foi dividido, sendo abordado, então, em 3 categorias, em que fatores base do Scrum foram adotados para tangenciar as seções de variáveis e perguntas a serem feitas aos entrevistados, para que dessa forma possa garantir uma análise mais assertiva sobre a aplicação da metodologia Scrum. O referencial teórico adotado foi o último “Guia do SCRUM” de 2020, elaborado em conjunto pelos criadores da metodologia Jeff Sutherland e Ken Schwaber.

Posto isso, o primeiro fator base se refere aos pilares do Scrum. O Guia do Scrum, cita que a metodologia combina quatro eventos formais para a inspeção e adaptação contidos dentro do evento Sprint, o qual a transparência dos artefatos permite a inspeção. Esses eventos funcionam por implementar os pilares empíricos do Scrum: transparência, inspeção e adaptação (Schwaber; Sutherland, 2020). Para inferir que os pilares estão sendo atendidos em sua modalidade remota, é imprescindível analisar se a forma que seus artefatos, eventos e a adaptação do trabalho estão sendo aplicadas minimamente da forma descrita em seu Guia. Desse modo, a análise das perguntas contidas nessa categoria visa responder se os eventos e artefatos estão sendo implementados da maneira adequada e o trabalho sofrendo adaptações quando necessário.

Por conseguinte, o segundo fator base são os valores pregados pela metodologia. O sucesso Scrum se encontra plenamente dependente das pessoas do

time viver os cinco valores: compromisso, respeito, coragem, foco e abertura (Schwaber; Sutherland, 2020). Sendo assim, as perguntas desta seção têm como objetivo responder se esses valores estão presentes no time investigado.

Por último, o terceiro fator base do adotado é o time Scrum. O time deve ser multifuncional e multidisciplinar, provendo assim todas as habilidades necessárias para realizar a entrega de valor ao final de cada *sprint*. As variáveis adotadas para identificar se o time Scrum analisado está de acordo com o guia, foram: composição do time Scrum e papéis e responsabilidades dentro do time.

As entrevistas foram realizadas via *Google Meet* e duraram em média 40 minutos cada. Ao iniciar, todos os entrevistados foram informados do conteúdo e finalidade da entrevista, e também que o intuito era primordialmente coletar as perspectivas dos mesmos acerca das suas vivências no dia a dia de seus projetos ao que tange a forma em que é gerenciado. As perguntas não foram compartilhadas com os entrevistados, mas o autor seguiu o roteiro do apêndice A.

Além disso, antes de rodar as entrevistas com os membros do projeto a ser utilizado como base para esta pesquisa, o autor executou um teste piloto com dois membros de outra instituição que adotava o Scrum também de forma remota. O intuito do teste piloto foi garantir que o ritmo das perguntas não seria cansativo e que a forma que foram elaboradas de fato retornaria respostas descritivas e com conteúdo relevante para o objetivo do trabalho. Ao final do teste piloto algumas perguntas foram reestruturadas.

### **3.4 Análise em relação ao time Scrum**

O time Scrum, de acordo com Schwaber e Sutherland (2020), deve ser pequeno com até 10 pessoas para se manter ágil, com a comunicação fluída e com o entendimento geral do que cada um está desenvolvendo, mas grande o suficiente para ser possível incrementos de valor ao final de cada *sprint*. O time é responsável por todas as atividades relacionadas ao produto e devem se manter auto gerenciáveis. Ainda segundo o guia do Scrum, o time deve contar com os papéis de *Scrum Master*, *Product Owner* e desenvolvedores.

O primeiro questionamento a todos os entrevistados foi quantas pessoas fazem parte do time e em quais papéis atuam. Todos confirmaram a presença de 1 SM, 1 PO e 4 devs que compõem a equipe. O PO chegou a citar:

Houve uma rotatividade significativa de desenvolvedores durante este tempo em que estou como PO deste projeto, principalmente por conta do mercado aquecido de TI, mas rapidamente as vagas foram preenchidas e o mínimo de devs que tivemos no time foi de 3 e o máximo 5, no momento em que precisamos de um pouco mais de braço para desenvolvermos algumas atividades, mas a equipe nunca passou de 7 pessoas (Product Owner).

Quando perguntado ao PO se esses desenvolvedores foram atuar em empresas conterrâneas, ele informou que todos foram para empresas espalhadas pelo Brasil, inclusive um deles foi atuar em uma empresa no Canadá, onde todo o processo seletivo aconteceu de forma remota e o mesmo atuará sem ter que sair do Ceará. Com essa informação, é possível inferir que o time se manteve dentro do limite aceitável de membros, mesmo com a facilidade encontrada para troca de empresas, considerando além do estado do instituto Ômega.

Ademais, os entrevistados foram questionados sobre as responsabilidades de cada papel no time, dando enfoque para as responsabilidades referente ao seu papel e em seguida solicitado que levantasse algumas das suas principais dificuldades para executá-las. Com relação a percepção de cada um sobre as responsabilidades do time, podemos observar no quadro abaixo (quadro 3) uma uniformidade nas visões de cada entrevistado sobre os demais membros do time, o qual o time de desenvolvimento atua no desenvolvimento em si do produto, o PO atua no gerenciamento de atividades e de como esse produto será entregue, de forma a alinhar os requisitos junto ao cliente e o SM atua retirando impedimentos do time, garantindo um bom funcionamento do Scrum e gerenciando pessoas. É possível destacar que o PO, por vezes, é visto pelo time como o SM “substituto”, fator que se deve a no dia a dia estar mais próximo de atividades do SM que do time de desenvolvimento e por estar na linha de frente do projeto ao alinhar requisitos com o cliente.

Tabela 3 - Visão sobre cada papel do time

	Visão dev1	Visão dev2	Visão dev3	Visão dev4	Visão PO	Visão SM
Papel desempenhado pelo PO	Faz um pouco do papel de SM removendo impedimentos junto ao cliente e garantindo a entrega das atividades.	Dar visão ao time de desenvolvimento sobre próximos passos e orientar o time para os objetivos a serem alcançados. Entender o problema do cliente e repassa-lo com clareza para o time de desenvolvimento.	Ajuda o time a definir atividades e suas prioridades no projeto. Assume por vezes o papel de SM, desbloqueando o time.	Gerencia backlog e alinha requisitos com o cliente.	Promove alinhamentos de produto com stakeholders e time, trabalha na priorização de itens a serem entregues, facilita algumas cerimônias scrum, elucida requisitos de negócio para funcionalidades a serem desenvolvidas.	Alinhamento de negócio com o cliente, gerencia o backlog e entregas da equipe de desenvolvimento.
Papel desempenhado pelo SM	Remove impedimentos.	Ponto de contato entre equipe contratada e equipe contratante, removendo impedimentos mais "graves". Facilitar comunicação com clientes.	Desbloquear problemas menos técnicos, os escalando para pessoas responsáveis.	Gerenciar equipe geral e alinhamentos específicos com o cliente.	Remove impedimentos do time, atua como gestor de pessoas garantindo um ambiente agradável no time e da suporte em alinhamentos com o cliente.	Principalmente garantir que o scrum seja seguido, sendo observador se o time está seguindo as cerimônias da melhor maneira. Faz gestão funcional: gestão de pessoas: se estão motivadas ou impedidas. Intervém quando percebe oportunidades com stakeholder ou time para melhorar processos e/ou produto.
Papel desempenhado pelos Devs	Codificação e priorização das atividades pelas quais é responsável pós planning.	Desenvolver sistema baseado nos requisitos trazidos pelo PO e pelo cliente.	Atuação na área técnica: desenvolvimento do sistema, correção de bugs, análise de ferramentas e soluções para contribuir com a qualidade do projeto. Testar aplicações desenvolvidas.	Desenvolver o produto tecnicamente e validar todas as funcionalidades desenvolvidas.	Desenvolvem o sistema descrito, testam o sistema desenvolvido, garantem qualidade do produto, promovem alinhamentos técnicos, estimam atividades.	Codificação do produto.

Fonte: Proposto pelo autor

É possível evidenciar que os papéis e responsabilidades citados pelos entrevistados condiz com o esperado segundo a literatura utilizada no referencial bibliográfico deste trabalho.

Tendo em vista tais aspectos, quando perguntados sobre as maiores dificuldades encontradas no dia a dia para desempenhar suas atividades de forma remota, o dev1 expôs que existe algumas dificuldades acerca de permissões para desenvolver algumas funcionalidades do projeto por conta de diretivas da empresa contratante, mas não vê nenhum empecilho ou dificuldade em desempenhar suas atividades de forma remota por ter muitos canais para ter acesso ao time e ao cliente. O dev3 levanta a mesma dificuldade acerca de autorização e acessos, e comenta que uma dificuldade constante é demora na resposta do cliente por conta da comunicação assíncrona, porém não chega a ser impeditivo. O dev2 cita como dificuldade o retrabalho em algumas funcionalidades após a mudança de requisitos pelo cliente, todavia não julga ser uma falha na comunicação ou causada pelo projeto ser remoto. Na visão do dev4, a principal dificuldade é o atraso em algumas atividades de teste por conta de aviso tardio do responsável pela atividade em

comunicar sua finalização. Ademais, para o PO, a falta de tempo para trabalhos mais operacionais, como atualização de planilhas e documentos e até mesmo para estruturação de ideias para debater com as partes interessadas, por conta das constantes reuniões ao longo do dia, foi a principal dificuldade na execução remota de suas atividades. Já para o SM, ele não sente dificuldades em desempenhar suas atividades e responsabilidades de forma remota.

Sempre acreditei no trabalho remoto, sempre achei que era tendência e que o trabalho presencial, pelo menos para a área de TI, seja um pouco ultrapassado, até mesmo pela facilidade de adquirir conhecimento na área. Dá sim para se comunicar com as pessoas, há maneiras de se fazer isso, e esse é o desafio: a dinâmica de inserir assuntos, a maneira que você fala e como quebrar o gelo. De forma presencial isso seria mais simples, mas hoje com a experiência vejo que esse desafio foi superado por mim para desenvolver meu papel. As pessoas trabalham, as pessoas produzem e não precisam de micro gerenciamentos para isso acontecer (Scrum Master).

Dessa forma, é conclusivo que os principais pontos de dificuldade no dia a dia do time Scrum do projeto refere-se à comunicação assíncrona para elucidação de itens junto ao cliente, ao alinhamento interno do time quanto à liberação de atividades para teste, e excessivo número de reuniões ao longo do dia comprometendo atividades operacionais do PO. Contudo, de forma geral, a estruturação do time Scrum se mantém fiel ao pregado pelo Guia do Scrum.

### **3.5 Análise em relação aos pilares Scrum**

Segundo Schwaber e Sutherland (2020), a *sprint* é um marcador de ciclos de desenvolvimento e contém todos os outros eventos da metodologia, onde cada um destes é uma oportunidade formal para inspecionar e adaptar os artefatos do Scrum permitindo a transparência necessária. É possível, então, ao analisar a forma de execução dos eventos do Scrum, analisar os pilares da metodologia: transparência, inspeção e adaptação.

Ao perguntar sobre os eventos do Scrum e como ocorrem no objeto de estudo desta pesquisa, os entrevistados responderam igualmente que a *sprint* ocorre com duração de duas semanas, o primeiro evento é o planejamento da *sprint* que dura em média duas horas e meia e, a partir daí, todos os dias o time se encontra nas reuniões diárias, via *Google Meet* por 15 minutos onde cada membro dá o seu reporte sobre o andamento de suas atividades, se há impedimento e o que

fará em seguida. O evento seguinte é a revisão da *sprint* com duração de aproximadamente 30 minutos, a qual ocorre a apresentação dos resultados alcançados durante aquela *sprint* para o cliente, de maneira que este dará o “aceite”, que significa que a *sprint* foi aceita e o incremento entregue gerou valor ao produto final na visão do cliente. Esta cerimônia ocorre via *Microsoft Teams*. Ademais, a última cerimônia executada pelo time é a retrospectiva da *sprint*, que ocorre ao final de cada duas *sprints* e tem como objetivo a revisão do processo implementado e a dinâmica do time desde a última retrospectiva, visando melhoria de processos e ajustes necessários. Esta reunião ocorre via *Google Meet*, tem duração de aproximadamente 1 hora e conta com o suporte do aplicativo Metro Retro para facilitar o evento.

### **3.5.1 Backlog do Produto**

De fato, as funcionalidades a serem implementadas em cada produto são gerenciadas em uma lista chamada backlog do produto. Para Wazlawick (2013), um dos princípios do manifesto ágil é usado aqui: adaptação em vez de planejamento, dessa forma, o backlog do produto não precisa ser completo no início do projeto. Pode-se iniciar esta lista apenas com as funcionalidades mais evidentes para, depois, à medida que o projeto avançar, descrever novas funcionalidades que forem sendo descobertas.

Quando questionado ao PO como o backlog do produto é estruturado e como a meta do produto é definida, ele afirma que a definição da meta do produto precede a criação do backlog. Por meio de reuniões iniciais junto ao cliente, é entendido quais as dores que ele busca tratar com aquele novo produto a ser desenvolvido e, a partir daí, então, será desenhada uma solução junto com time Scrum para dar uma ideia geral de como aquele produto ficará para atingir os objetivos esperados. Por conseguinte, ao validar isso com o cliente, a meta do produto é definida e norteará o que deverá ser feito para entrega da solução esboçada, ou seja, em seguida é realizado levantamento de itens que comporão o backlog do produto. Ao ser questionado qual a dificuldade sente nesse processo sendo feito de forma remota, o PO esclarece:

Sem dúvidas transmitir a mensagem e as soluções pensadas de forma clara tanto para o time quanto para o cliente. Por vezes as reuniões em que

discutimos isso são exaustivas para garantir que todos expuseram seus pontos de vista e garantirmos que todos os envolvidos estão alinhados quanto ao que será entregue e desenvolvido. Depois de um tempo, passei a utilizar o Miro como forma de apresentar visualmente o que estava querendo passar por meio de fluxograma e esboço de tela, com isso percebi uma grande evolução no entendimento dos cenários discutidos. Apesar de levar um pouco mais de tempo para estruturar esses momentos, saio com a sensação de maior conforto do que quando discutimos apenas falando (Product Owner).

Como explicação, o Miro é uma lousa digital utilizada para ajudar na colaboração com outras pessoas de forma online, a qualquer momento, em qualquer lugar e funciona como um quadro branco infinito e interativo.

Ademais, os outros membros do time confirmaram em suas entrevistas que a meta do produto é clara para todos, em especial o SM citou que, quando necessário, ajuda no esclarecimento deste junto ao cliente e ao time. Com relação ao gerenciamento e acompanhamento do backlog, o PO comenta que o mantém por meio do *Azure DevOps*, uma ferramenta online que permite a criação, ordenação e quebra em subatividades para funcionalidades e itens criados. Assim, atende as necessidades do time principalmente por permitir uma definição de *sprints* e, com isso, registrar o que foi trabalhado em cada uma delas, quem foi o responsável e o tempo que levou.

Durante o reporte dado nas *dailies* (reuniões diárias), o PO compartilha a tela com o time e apresenta o quadro de atividades no *Azure DevOps* e cada membro fala como está o andamento das suas atividades. Ao atualizar o quadro de atividades o time todo fica ciente de em que estado está sua atividade sem a necessidade de reunião ou perguntas (dev1).

### **3.5.2 Eventos**

O planejamento da *sprint* definirá o trabalho a ser desenvolvido durante a *sprint* e este plano resultante é criado pelo trabalho colaborativo de todo o time Scrum (Schwaber, Sutherland, 2020). O PO garante que os itens mais importantes e priorizados do backlog do produto estejam claros e descritos para facilitar a discussão do time e como eles são mapeados para atingir a meta do produto.

A reunião acontece via *Google Meet* a cada início de *sprint*, ou seja, a cada 2 semanas para este time. Neste momento, o time irá discutir o que será entregue ao final daquela *sprint*, quem o fará e como o fará. Sendo assim, na visão

de todo o time de desenvolvimento, essa é a cerimônia mais complicada do Scrum, principalmente por ser extensa e necessitar da atenção de todos naquele momento para não causar desalinhamento ao longo da *sprint* e principalmente para garantir que o que está sendo definido será entregue.

Além disso, para definir o que é possível realizar dentro da *sprint*, o time do instituto Ômega utiliza a estimativa de pontos para as tarefas (*story points*). De acordo com Rodrigo Pinto (2021), o *story point* é unidade de complexidade de esforço que se tem para construir um pedaço de software que não segue uma escala linear, e passa a usar uma escala exponencial, quadrática, logarítmica ou Fibonacci. Mediante tais aspectos e para não enviesar a pontuação de uma determinada atividade, o time analisado também adota a utilização do *planning poker*, uma ferramenta que garante que não exista uma polarização da estimativa, na qual, após a definição da escala a ser utilizada, no caso do time estudado a série de Fibonacci, durante os planejamentos da *sprint*, o PO dá uma descrição do que será a atividade, o time de desenvolvimento tira dúvidas se houver e então passa para a estimativa, que utilizando o *planning poker* funciona da seguinte forma: cada membro entra em uma sala do site *agilepoker.app*, da sua estimativa de forma individual e privada, e quando todos do time finalizam a sua estimativa, as cartas com o valor da estimativa (baseados na série de Fibonacci) de cada membro são reveladas e então se analisa como o item foi estimado por cada um, e no caso de haver grandes divergências, discussões são levantadas para entender a complexidade vista por cada um para exercer a atividade, até que, em seguida o time chega em um consenso e a estimativa é definida para aquele item.

Para o dev1, durante este projeto foi seu primeiro contato com a utilização de *story points* e *planning poker*, pois nas outras empresas em que trabalhou utilizava horas de desenvolvimento como estimativa, e o mesmo julga que estimar em horas apesar de parecer mais concreto, a utilização de *story points* se torna melhor e mais assertiva com relação ao que cabe em uma *sprint*.

Na visão do PO, a adoção de *story points* garantiu um maior conforto ao time para estimar atividades que antes eram estimadas em horas. Para ele, ter que ser exato com as horas de desenvolvimento gerava um efeito no time de estimar “para cima” cada item, para não se comprometer com algo e acabar não entregando ao final da *sprint*, após a adoção dos *story points* isso ficou de lado e o time vem sendo mais assertivo com suas estimativas, logo garante que não haverá grandes

movimentações de atividades puxadas do backlog do produto no meio da *sprint*, o que ocorria quando alguém estima 20 horas para uma atividade mas levou 8h e também garante também poucos atrasos nas entregas ao diminuir a passagem de uma atividade alocada na *sprint X* para a *sprint X+1* por ter estimado menos horas que o necessário.

Com relação a duração e desgaste do planejamento da *sprint* como um todo, o time foi unânime em citar que esse é o evento mais impactado pela adaptação no formato remoto. O dev2 cita que o debate do escopo das atividades sofre com a modalidade remota, principalmente por um esperar o outro finalizar a fala para então começar a falar e dessa forma alongar ainda mais a sua duração. O dev3 concorda com a colocação do dev2, ao citar que naturalmente por ter muitas etapas o planejamento da *sprint* se torna longo, mas principalmente por conta de debates acerca dos itens que irão compor a *sprint*, debate este que pode se alongar por minutos por item, mas não pode afirmar que se ocorresse de forma presencial seria diferente. Dev1 e dev4 citam haver um desconforto com a duração do momento, principalmente para manter o foco em todos os itens discutidos, todavia não visualizam melhorias do evento acontecer remotamente. Para o PO, a principal queixa é garantir que todos estão focados no que está sendo discutido para evitar repetições e desalinhamento ao longo da *sprint*, ele fala que:

É comum que ao longo da *sprint* alguns desenvolvedores me procurem para maiores detalhamentos acerca das suas atividades, principalmente se elas foram discutidas no final do planejamento. Por mais que eu deixe bem descrito o que deverá ser atingido, gera um maior conforto em ter um “mini planejamento” com o responsável pela atividade quando o mesmo for desenvolvê-la durante a *sprint*, pois gera um entendimento mais detalhado e evita retrabalho. Algumas vezes até opto por não dar tantos detalhes e passar regra a regra nas discussões da *planning* por fazer a leitura que isso dificultaria ainda mais o foco no que realmente importa. Acredito que fazer pausas ao longo do evento e realizarmos o evento de câmera ligada possa gerar um maior foco no que está sendo discutido e diminuição em atividades paralelas durante o momento (Product Owner).

O Dev4 cita um momento que evidencia o que foi descrito acima, quando ocorreu de durante a reunião diária um membro indagar o porquê daquela atividade estar sendo desenvolvida, quando a mesma havia sido debatida e alinhada durante o planejamento.

É inevitável a perda de atenção em uma reunião de 2 horas onde tudo é importante (Dev4).

Indo de encontro com as percepções do restante do time, o SM revela que acredita que o evento de planejamento está mais objetivo que quando ocorria presencialmente e que nesta modalidade duravam um turno todo, às vezes chegando a ser o dia inteiro de planejamento, mas concorda que esse objetivismo tem seu preço, tornando, assim, sessões mais intensas e mais cansativas. Ele também cita que acredita que de forma presencial as discussões sobre os itens eram mais extensas, mas que, por vezes, perdiam o foco.

Para guiar o planejamento da *sprint*, o time também define a meta daquele ciclo baseado na definição de pronto, acordado previamente pelo time. Ambos são definidos em conjunto com todo o time e todos os entrevistados citam que todos possuem ciência desses pontos.

Adiante, o anexo A apresenta a tabela utilizada pelo time do instituto Ômega para guiar as estimativas de cada item a ser trabalhado. Para o time, cada membro deve ser responsável por no máximo 13 *story points*, sejam eles alocados em um único item ou em diversos itens. Para o time também foi decidido que nenhum item poderia ter mais de 13 *story points*, pois é entendido que ao passar dessa pontuação o item pode ser mais esclarecido quanto a requisitos ou quebrado em atividades menos complexas.

Quando indagados acerca de quais outros eventos foram mais impactados pelo remoto, a maioria citou que não via outros impactos, mas podemos citar duas percepções que se destacaram. Com relação a *sprint review*, tanto o dev1 como o SM levantam pontos relacionados a condução desses momentos:

Nas apresentações para o cliente, não consigo sentir uma interação tão grande como acredito que seria se fosse de forma presencial. Pode ser relacionado ao fato do cliente ser um pouco mais distante, mas mesmo assim quando vamos apresentar os resultados, não conseguimos observar as reações, movimentações e etc. enquanto apresentamos, e isso me deixa com a sensação de estar no escuro até que toda a apresentação finalize eu possa ouvir os feedbacks. Se a reunião ocorresse com as câmeras ligadas talvez essa sensação diminuiria um pouco (dev1).

Já o SM revela que antes da pandemia, ocorriam marcos do projeto em que o time Scrum que trabalhava entre si de forma presencial encontrava o cliente

também de forma presencial, e estes contatos ainda que pontuais, estreitaram as relações do time com o cliente, gerando mais empatia.

Um aperto de mão durante algumas etapas do projeto faz com que o time veja aquele produto com uma cara e não só como um entregável qualquer. Sinto que gera uma motivação maior de ambas as partes em fazer dar certo (Scrum Master).

Outro evento citado foi a retrospectiva da *sprint*, para o SM ela não está cumprindo totalmente o objetivo que deveria cumprir. Segundo o Guia do Scrum, a retrospectiva existe para que o time inspecione como foi a última *sprint* em relação a indivíduos, interações, processos, ferramentas e sua definição de pronto. Após as melhorias serem levantadas, o time identifica as mudanças mais úteis para melhorar sua eficácia.

Sinto que a retrospectiva funciona mais como um momento de agradecer as pessoas do time de alguma forma, levantar alguns problemas encontrados e identificar melhorias no dia a dia de acordo com as últimas duas sprints, mas para estes dois últimos vejo muita pouca ação ser tomada e de fato posta em prática (Scrum Master).

O time de desenvolvimento não levantou nenhuma dificuldade ou ponto de atenção na execução da retrospectiva da *sprint*, já o PO comentou que acredita que o espaçamento de acontecer apenas de duas em duas *sprints* dificulta lembrar tudo que aconteceu e pontos de incômodo a serem melhorados.

Hoje utilizamos o Metro Retro que é uma lousa digital assim como o Miro, para conduzir a sprint retrospectiva. Creio que a forma em que é estruturada não é tão voltada para a identificação de pontos de melhorias e acaba sendo um momento de desabafo geral do time. Não chegamos de fato a revisar os processos adotados e trabalhar em cima de melhorias neles. Fica tudo muito subjetivo (Product Owner).

Todo o time citou que não há nada imutável no processo adotado e no produto sendo desenvolvido que não possa ser adaptado se necessário, assim como não há uma resistência do time caso isso não seja evitável.

É possível concluir que a retrospectiva da *sprint* foge um pouco do modelo sugerido pelo Guia do Scrum, onde deveria ser realizada ao final de cada *sprint*, mas não é possível inferir que a percepção do time com relação a sua efetividade está relacionada com ela estar sendo aplicada no modelo remoto. Para a revisão da *sprint* apesar dos pontos levantados acerca do distanciamento com o cliente, não gerou insumo suficiente para invalidar ou comprometer o funcionamento

dela no modelo remoto. O mesmo acontece com o evento de planejamento, onde algumas ações podem ser tomadas para deixá-las menos cansativas, como sugerido pelo PO, mas não há evidências de um mau funcionamento da forma em que está sendo aplicada. É evidente que a partir da coleta de dados na entrevista, os pilares Scrum vem sendo atendidos no time. Na Tabela abaixo há uma relação de ferramentas que auxiliam o time na execução dos eventos e artefatos de forma remota.

Tabela 4 - Ferramentas de auxílio para reuniões remotas

<b>Evento / Arteto Scrum</b>	<b>Ferramentas de suporte</b>
Planejamento de sprint	<i>Google Meet, Miro, AgilePoker, Google Sheets</i>
Reuniões diárias	Google Meet, Azure DevOps
Revisão de sprint	<i>Microsoft Teams, Google Apresentações</i>
Retrospectiva de sprint	Google Meet, Metro Retro
Gerenciamento de Backlog	Azure DevOps

Fonte: Proposto pelo autor

### 3.6 Análise em relação aos valores Scrum

O Guia do Scrum, fala que os 5 valores do time Scrum são: compromisso, foco, abertura, respeito e coragem. Diante disso, o time deve se comprometer a atingir seus objetivos, tendo em vista que o foco principal é o trabalho da *sprint*, além de dever ser aberto quanto ao trabalho e os desafios. Ademais, devem se respeitar quanto a serem pessoas capazes e independentes, e ter a coragem de fazer a coisa certa e trabalhar em problemas difíceis (Schwaber; Sutherland, 2020).

Dessa maneira, ao questionar aos entrevistados se durante o período pandêmico eles despendem tempo em atividades ligadas de alguma forma ao trabalho remoto que pudessem comprometer seu foco no trabalho a ser feito, unanimemente todos responderam que não ocorreu nada que de fato comprometesse suas atividades e, quanto ao foco, o dev3 comentou que em

alguns momentos em que seu vizinho estava reformando a casa, o barulho oriundo de furadeiras e martelos atrapalhou um pouco o foco, mas nada preocupante. Já na visão de dev4, o principal fator que pode retirar seu foco de desenvolvimento é quando ocorrem muitas reuniões durante o dia, o deixando praticamente sem codificar.

Com relação ao comprometimento, abertura, respeito e coragem, ficou muito evidente na fala de todos o quanto o time era integrado e comprometido em ajudar uns aos outros. Na visão do dev1, isso é fortalecido pelos desafios diários da solução técnica proposta no projeto, onde é necessário que se haja muito alinhamento e sinergia da equipe. O dev3 comenta que em determinadas situações em que é necessário realizar uma “força-tarefa” para resolução de alguma dificuldade de algum dos membros, o time todo se mostra disponível e entram em reuniões para chegarem juntos a uma solução sem muita demora na prontidão. O dev4 cita que o clima de pedir ajuda é sempre frisado em qualquer oportunidade e isso contribui para um ambiente de trabalho aberto a discussões.

De resto, para o PO, a proatividade de todo o time contribui para que os problemas sejam rapidamente debatidos e contornados e devido, sobretudo, o cliente ser externo ao instituto Ômega, e alguns dos alinhamentos serem feitos em inglês, fica ainda mais claro a colaboração do time e o empenho para contornar as dificuldades encontradas. Além disso, na visão do SM, todo o trabalho institucional de fortalecimento da cultura realizado no instituto Ômega contribui para que o ambiente colaborativo seja facilmente transmitido para os projetos e seus membros.

Dessa maneira, verifica-se possível concluir que o time analisado mantém vivo os valores Scrum ainda que de forma remota.

#### **4 CONCLUSÃO E TRABALHOS FUTUROS**

O presente trabalho se propôs a analisar a aplicação da metodologia Scrum de forma remota, bem como identificar dificuldades e benefícios nesta aplicação. É possível concluir que o objetivo foi alcançado e comprovou a aplicabilidade da metodologia de forma remota sem perdas significativas para o ambiente analisado.

Ao questionar o time estudado sobre a visão dos mesmos acerca da possibilidade do Scrum correr bem de forma remota, foi unânime a resposta de que sim era possível, e que de acordo com a experiência deles a metodologia rodava tão bem quanto rodaria de forma presencial.

Por fim, em relação às sugestões para trabalhos futuros, propõem-se:

a) Realizar uma análise comparativa entre a aplicação da metodologia Scrum de forma remota e presencial em um mesmo nicho de projeto;

b) Analisar a aplicação da metodologia Scrum de forma remota para outro tipo de projeto que não seja de software.

## REFERÊNCIAS

ALMEIDA, Guilherme Augusto Machado de. **Fatores de escolha entre metodologias de desenvolvimento de software tradicionais e ágeis**. 2017. Tese de Doutorado. Universidade de São Paulo.

BARCELLOS, Monalessa Perini; ROCHA, Ana Regina; TRAVASSOS, Guilherme Horta. **Planejamento de Custos em Ambientes de Desenvolvimento de Software Orientados à Organização**. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE.

BARROS, Alexandre Moço; SILVA, José Roberto Gomes da. **Percepções dos indivíduos sobre as consequências do teletrabalho na configuração homeoffice: estudo de caso na Shell Brasil**. CADERNOS Ebape. br, v. 8, p. 71-91, 2010.

BECK, K. et al. **Manifesto para desenvolvimento ágil de software**. 2001. Disponível em: <http://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 20 jan. 2022

BIGARELLI, Barbara. **43% das empresas adotam home office devido ao coronavírus: pesquisa com 359 companhias mostra que trabalho remoto é um desafio**, 2020. Disponível em: <https://valor.globo.com/carreira/noticia/2020/03/20/43percent-das-empresas-adotamhome-office-devido-ao-coronavirus.ghtml>. Acesso em: 21 jan. 2022..

BRIDI, Maria Aparecida et al. **O trabalho remoto/home-office no contexto da pandemia COVID-19**. Curitiba: Universidade Federal do Paraná, Grupo de Estudos Trabalho e Sociedade, 2020.

CIO. **Home office e teletrabalho são cada vez mais comuns no Brasil**, 2016. Disponível em: <https://cio.com.br/gestao/home-office-e-teletrabalho-sao-cada-vez-mais-comuns-no-brasil/>. Acesso em: 23 jan. 2022.

DIGITAL.AI. **15th Annual State Of Agile Report**, 2020. Disponível em: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>. Acesso em: 03 jan. 2022

DOS SANTOS CARVALHO, Francisco et al. **Práticas ágeis para gestão de projetos baseadas no PMBOK guide, no agile practice guide e na metodologia scrum: uma análise da aplicabilidade em um projeto-piloto de uma organização pública**. Revista de Ciência da Computação, v. 2, n. 1, p. 53-60, 2020.

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

FADEL, A. C; SILVEIRA, H. M. **Metodologias ágeis no contexto de desenvolvimento de software: XP, SCRUM e Lean**. 2010. 26 p. Dissertação (Mestrado em Tecnologia) - Faculdade de Tecnologia, Universidade Estadual de Campinas, Limeira, 2010.

FISCHER, Andre Luiz. **Um resgate conceitual e histórico dos modelos de gestão de pessoas**. In. As Pessoas nas Organizações. Ed. Gente, 2002.

GODOY, A. S. **Pesquisa qualitativa: tipos fundamentais**. Revista de Administração de Empresas, v. 35, n. 3, p. 20-29, 1995.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. 5. ed.-São Paulo: Atlas, 2003.

PAULA FILHO, W. P. **Engenharia de software**. LTC, 2003.

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

PINTO, Rodrigo. **Story Points: Entenda de uma vez por todas como fazer estimativas**, 2021 Disponível em: <https://agileschool.com.br/story-points-entenda->

de-uma-vez-por-todas-como-fazer-estimativas/. Acesso em: 23 jan. 2022. RIBEIRO, S. A.; SCHMITZ, E. A.; ALENCAR, A. J. S. M.; SILVA, F.. **A Síndrome do Deadline: origem, causas e implicações no processo de desenvolvimento de software.** Revista Brasileira de Sistemas de Informação, v. 10, n. 2, p. 30-47, 2017.

ROYCE, W. **Managing the development of large software systems: Concepts and techniques.** In: Proc. IEEE WESCOM. IEEE Computer Society Press, Los Alamitos. 1970.

SCHNELL, Alexandre. **Uso da modelo cascata no desenvolvimento de um sistema para serviços de oficina mecânica automotiva.** 2018. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

SCHWAB, Klaus. **A quarta revolução industrial.** Edipro, 2019.

SCHWABER, K.; SUTHERLAND, J. Guia SCRUM. [S. l.]: **Scrum Guides, 2020. Ebook.** 19 p. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-2.0.pdf>. Acesso em: 23 jan. 2022.

SINGLENTON, Jr. Royce et alli. **Aproches to social research.** New York: Oxford University Press, 1970.

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

STANKIEWICZ, Alessandro. **Modelo de interação ágil: uma adaptação da modelo cascata à organização de pequenas e médias empresas.** 2017. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

SUTHERLAND, J. **SCRUM: a arte de fazer o dobro do trabalho na metade do tempo.** Tradução: Nina Luna. 2. ed. São Paulo: Leya, 2016.

WAZLAWICK, R. **Engenharia de software: conceitos e práticas.** Rio de Janeiro: Elsevier, 2013.

WORLD ECONOMIC FORUM, **The Future of Jobs Report 2020**, 2020. Disponível em: <https://www.weforum.org/reports/the-future-of-jobs-report-2020>. Acesso em: 03 jan. 2022

YIN, R K. **Estudo de caso: planejamento e métodos**. 2ed. Porto Alegre: Bookman, 2001.

ZEETANO CHAHAD, J. P. **O futuro do trabalho pós Covid-19**. Ciência & Trópico, [S. l.], v. 45, n. 1, 2021. DOI: 10.33148/cetropicov45n1(2021)art6. Disponível em: <https://periodicos.fundaj.gov.br/CIC/article/view/1988>. Acesso em: 30 jan. 2022.

## ANEXO A – MATRIZ DE STORY POINTS

MATRIZ STORY POINTS		Incerteza				
		Trivial	Baixa	Média	Alta	Muito alta
Complexidade	Trivial	1	2	3	5	8
	Baixa	2	3	5	8	13
	Média	3	5	8	13	21
	Alta	5	8	13	21	34
	Muito alta	8	13	21	34	100

Fonte: Produzido pelo time entrevistado

## APÊNDICE A – PERGUNTAS UTILIZADAS NAS ENTREVISTAS

<b>Time Scrum</b>
Quantas pessoas fazem parte do seu time e quais papéis desempenham? (pode ser detalhista)
Quais dificuldades são mais comuns nas tuas atividades do dia a dia e quais delas são devido a adoção do trabalho remoto?
<b>Pilares Scrum</b>
Como e em que ordem ocorrem os eventos Scrum?
Quais ferramentas são utilizadas para facilitar cada evento?
Quais as principais dificuldades na execução remota de cada uma?
Nos momentos em que há dúvidas ou necessidade de esclarecimentos sobre alguma atividade quais as principais dificuldades encontradas?
Como é construído o backlog do produto em que trabalham?
Já aconteceu alguma situação em que o time teve que se adaptar de alguma forma no projeto?
<b>Valores Scrum</b>
Durante o trabalho remoto você percebeu algum membro do time descomprometido com as metas traçadas em conjunto?
Caso um membro do time encontre algum tipo de dificuldade em desempenhar suas atividades, como isso é contornado?
Você considera que o ambiente em que trabalha é propício para tirar dúvidas, expor opiniões, levantar discussões e etc?
Durante alguma sprint você desenvolveu atividades paralelas às planejadas durante o evento da planning que comprometeram sua dedicação nessas atividades planejadas?
<b>Outros</b>
Para você é possível, através de sua experiência, a utilização da metodologia Scrum em regime remoto?
Maior benefício e maior malefício de trabalhar com o Scrum em formato remoto?

Fonte: Produzido pelo autor