

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia de Software

***Requirements Sprint: Uma Abordagem  
Orientada a Requisitos de Software para  
Projetos Ágeis de Pequeno e Médio Portes***

Autor: Kamilla Costa Souza  
Orientador: Profa. Dra. Milene Serrano

Brasília, DF  
2021





Kamilla Costa Souza

***Requirements Sprint: Uma Abordagem Orientada a  
Requisitos de Software para Projetos Ágeis de Pequeno e  
Médio Portes***

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Profa. Dra. Milene Serrano

Coorientador: Prof. Dr. Maurício Serrano

Brasília, DF

2021

---

Kamilla Costa Souza

*Requirements Sprint: Uma Abordagem Orientada a Requisitos de Software para Projetos Ágeis de Pequeno e Médio Portes* / Kamilla Costa Souza. – Brasília, DF, 2021- 143 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Milene Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB

Faculdade UnB Gama - FGA , 2021.

1. Abordagem. 2. Requirements Sprint. 3. Ágil. 4. Elicitação. 5. Modelagem. 6. Análise. 7. Requisitos. I. Profa. Dra. Milene Serrano. II. Universidade de Brasília. III. Faculdade UnB Gama. IV.

*Requirements Sprint: Uma Abordagem Orientada a Requisitos de Software para Projetos Ágeis de Pequeno e Médio Portes*

CDU 02:141:005.6

---

Kamilla Costa Souza

# ***Requirements Sprint: Uma Abordagem Orientada a Requisitos de Software para Projetos Ágeis de Pequeno e Médio Portes***

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 21 de Maio de 2021:

---

**Profa. Dra. Milene Serrano**  
Orientador

---

**Prof. Dr. Maurício Serrano**  
Coorientador

---

**Prof. Dr. André Barros de Sales**  
Convidado 1

Brasília, DF  
2021



*Este trabalho é dedicado aos homens da minha vida, que não estão mais presentes fisicamente, mas que sonharam com este momento tanto quanto eu, que se dedicaram e incentivaram ao máximo para que isso pudesse se tornar concreto, bem como para que eu me tornasse uma mulher independente. Com todo meu amor e gratidão, dedico este trabalho ao meu pai José Eduardo de Souza e Silva, ao meu tio Carlos Roberto da Costa Barbosa e aos meus avôs Francisco Barbosa de Sousa e José de Souza e Silva.*





# Agradecimentos

Agradeço, primeiramente, a Deus por cuidar tão bem de mim e por tantas bênçãos alcançadas como a oportunidade de cursar a faculdade na Universidade de Brasília, ter uma família que me apoia, amigos que me incentivam e orientadores que admiro.

Agradeço em especial aos meus pais, por todo esforço e dedicação de me proporcionar as melhores condições possíveis para meu desenvolvimento. Agradeço minha mãe, Maria do Perpétuo, por todo amor e sacrifício para que fosse possível estar realizando a faculdade; por não medir esforços para me ver feliz, e por me fazer querer ser uma grande mulher, assim como ela o é. Agradeço ao meu pai, José Eduardo, que mesmo não estando mais presente fisicamente, lutou toda a vida para me oferecer o melhor estudo e educação. Espero estar retribuindo toda alegria e todo orgulho que sinto por ter tido a oportunidade de tê-lo como pai nesta vida.

Agradeço à minha irmã e melhor amiga, Karolina Costa, que sempre esteve ao meu lado me apoiando, aconselhando e por ser o melhor presente com que Deus poderia me abençoar.

Agradeço à pessoa que amo e escolhi para partilhar minhas conquistas, Diego Alessandro, que sempre me apoiou durante momentos difíceis, que me incentivou a dar o meu melhor mesmo quando tudo parecia impossível, e a entender que as coisas tendem sempre a melhorar.

Agradeço à minha madrinha, Conceição, e ao meu tio Carlos, que sempre estiveram presentes em minha vida, cuidando e me apoiando.

Agradeço ao meu amigo, Lucas Vitor, por sua amizade, carinho e auxílio quando a caminhada se tornava mais árdua. Obrigada por tornar tudo mais leve e feliz.

Agradeço a todos os meus professores, que ao longo da graduação me incentivaram a ser melhor a cada desafio. Agradeço ao Professor Renato Coral por me abrir oportunidades, e por acreditar em meu potencial e em especial aos meus orientadores, Milene e Maurício, que me inspiram e por quem tenho grande admiração.

A todos estes, meu mais sincero muito obrigada.



*"Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota."(Madre Teresa de Calcutá)*



# Resumo

O processo de engenharia de requisitos, por mais necessário que seja para alcance do sucesso de projetos, ainda é mal definido, e por vezes inutilizável, fazendo com que muitas empresas/organizações, que estão inserindo produtos no mercado, tenham prejuízos altos em termos de custos e prazos de entrega. Um problema que estas empresas/organizações vivenciam é a inexistência de um processo de engenharia de requisitos bem definido, orientado às necessidades da empresa/organização quanto à elicitação, modelagem e análise de requisitos. Tal definição proporcionaria, dentre outras contribuições, rastreabilidade do gerenciamento interno; facilidade de comunicação entre clientes e empresa e, consequentemente, maiores chances de sucesso nas soluções computacionais desenvolvidas. Nesse trabalho, contribui-se com uma abordagem orientada a requisitos de *software* que permite tratar as atividades de elicitação, modelagem e análise de requisitos condizente com a necessidade de entregas rápidas. A abordagem estabelece um prazo de cinco dias para a realização dessas atividades, nos quais há envolvimento da equipe técnica e dos clientes. Conferiu-se o nome de *Requirements Sprint*, uma vez que a abordagem foi contruída com embasamento em três pilares: Engenharia de Requisitos, Metodologias Ágeis e *Design Sprint*.

**Palavras-chaves:** Abordagem - *Requirements Sprint* - Ágil - Elicitação - Modelagem - Análise - Requisitos.



# Abstract

The requirements engineering process, however necessary for the achievement of the success of projects, is still poorly defined, and sometimes unusable, making many companies/associations, which are certain products on the market, have losses high in terms of costs and delivery times. One problem that these companies/organizations experience is the lack of a well-defined requirements engineering process, oriented to the needs of the company/organization regarding the bidding, modeling and analysis of requirements. Such a definition would provide, among other contributions, traceability of internal management; ease of communication between customers and company and, consequently, greater chances of success in the computational solutions developed. In this work, we propose an approach oriented to software requirements that allows to treat the activities of bidding, modeling and analysis of requirements consistent with the need for fast deliveries. The approach establishes a period of five days for the realization of these activities, in which there is involvement of the technical team and customers. It was awarded the name of Requirements Sprint, since the approach was built on three pillars: Requirements Engineering, Agile Methodologies and Sprint Design.

**Key-words:** Approach - Requirements Sprint - Agile - Bidding - Modeling - Analysis - Requirements.





# Lista de ilustrações

Figura 1 – Contextualização da abordagem . . . . .	27
Figura 2 – Diagrama de causa e efeito ( <i>fishbone</i> ). . . . .	29
Figura 3 – Diagrama de processo de leitura. . . . .	32
Figura 4 – Classificação da metodologia segundo <i>Scrum</i> . . . . .	34
Figura 5 – <i>Timeline</i> referente às técnicas de elicitação de requisitos. . . . .	36
Figura 6 – Levantamento do método mais utilizado para Engenharia de Requisitos. . . . .	42
Figura 7 – Processo do <i>framework Scrum</i> . . . . .	44
Figura 8 – Utilização da plataforma <i>github</i> na monografia. . . . .	50
Figura 9 – Utilização da ferramenta <i>trello</i> para gerenciamento do projeto. . . . .	51
Figura 10 – Utilização do editor <i>overLeaf</i> na edição da monografia. . . . .	51
Figura 11 – Utilização do <i>zotero</i> para gerenciamento das referências desse trabalho. . . . .	52
Figura 12 – Ferramenta para elaboração de <i>wireframes</i> . . . . .	55
Figura 13 – Ferramenta para elaboração de <i>mockups</i> . . . . .	56
Figura 14 – Exemplo de <i>Rich Picture</i> . . . . .	57
Figura 15 – Ferramenta para elaboração de diagrama de sequência. . . . .	58
Figura 16 – Ferramenta <i>trello</i> para gerenciamento dos resultados de análise do protótipo. . . . .	58
Figura 17 – Ferramenta <i>zenhub</i> destinada ao gerenciamento do projeto . . . . .	59
Figura 18 – Fluxograma para um Estudo de Caso Múltiplo . . . . .	64
Figura 19 – Processo de condução do Trabalho de Conclusão de Curso 1. . . . .	65
Figura 20 – Processo de condução do Trabalho de Conclusão de Curso 2. . . . .	67
Figura 21 – Cronograma de atividades referente ao Trabalho de Conclusão de Curso 1. . . . .	69
Figura 22 – Cronograma de atividades referente ao Trabalho de Conclusão de Curso 2. . . . .	70
Figura 23 – Fluxo da abordagem <i>Requirements Sprint</i> . . . . .	73
Figura 24 – <i>Workflow</i> da abordagem <i>Requirements Sprint</i> . . . . .	76
Figura 25 – Lista de artefatos e justificativa de utilização . . . . .	90
Figura 26 – Questão 1 do Questionário de Avaliação . . . . .	102
Figura 27 – Questão 2 do Questionário de Avaliação . . . . .	103
Figura 28 – Questão 3 do Questionário de Avaliação . . . . .	103
Figura 29 – Questão 4 do Questionário de Avaliação . . . . .	104
Figura 30 – Questão 5 do Questionário de Avaliação . . . . .	104

Figura 31 – Questão 6 do Questionário de Avaliação . . . . .	105
Figura 32 – Questão 7 do Questionário de Avaliação . . . . .	105
Figura 33 – Questão 8 do Questionário de Avaliação . . . . .	106
Figura 34 – Questão 9 do Questionário de Avaliação . . . . .	107
Figura 35 – Questão 10 do Questionário de Avaliação . . . . .	107
Figura 36 – Questão 11 do Questionário de Avaliação . . . . .	107
Figura 37 – Esboço da abordagem preliminar . . . . .	124
Figura 38 – Fluxo da abordagem preliminar proposta . . . . .	126
Figura 39 – Resultado das práticas específicas aplicadas. . . . .	142
Figura 40 – Resultado das práticas genéricas aplicadas. . . . .	143

# Lista de tabelas

Tabela 1 – Tabela referente ao estudo de casos sobre técnicas de elicitação de requisitos em <i>startups</i> . . . . .	37
Tabela 2 – Tabela de especificação da utilização de técnicas de elicitação por <i>startups</i> . . . . .	39
Tabela 3 – Classificação da pesquisa . . . . .	63



# Lista de abreviaturas e siglas

ARE	Engenharia Ágil de Requisitos
CMMI	Modelo de Capacidade e Maturidade Integrado.
DT	<i>Design Thinking</i>
GV	Google Ventures
MVP	Produto Mínimo Viável
TRE	Engenharia de Requisitos Tradicional
PIX	Pagamento Instantâneo Brasileiro
XP	<i>Extreme Programming</i>



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>25</b>
<b>1.1</b>	<b>Contextualização</b>	<b>25</b>
<b>1.2</b>	<b>Questão de Pesquisa</b>	<b>27</b>
<b>1.3</b>	<b>Justificativa</b>	<b>27</b>
1.3.1	Revisão Sistemática	28
<b>1.4</b>	<b>Objetivo Geral</b>	<b>28</b>
1.4.1	Objetivos Específicos	29
<b>1.5</b>	<b>Organização do Trabalho</b>	<b>30</b>
<b>2</b>	<b>REFERÊNCIAL TEÓRICO</b>	<b>31</b>
<b>2.1</b>	<b>Considerações Iniciais</b>	<b>31</b>
<b>2.2</b>	<b>Engenharia de Requisitos</b>	<b>31</b>
2.2.1	Identificação do Estado Atual do Processo de Gestão de Requisitos	33
<b>2.3</b>	<b>Atividades dos Processos de Engenharia de Requisitos</b>	<b>35</b>
2.3.1	Atividade de Elicitação de Requisitos	35
2.3.2	Atividade de Modelagem de Requisitos	40
2.3.3	Atividade de Análise de Requisitos	40
<b>2.4</b>	<b>Métodos Ágeis</b>	<b>42</b>
2.4.1	Práticas Ágeis do Scrum	43
2.4.2	Projetos Ágeis de Pequeno e Médio Portes	44
2.4.3	<i>Design Sprint</i> - Método Base para a Abordagem Desenvolvida	46
<b>2.5</b>	<b>Resumo do Capítulo</b>	<b>48</b>
<b>3</b>	<b>SUPORTE TECNOLÓGICO</b>	<b>49</b>
<b>3.1</b>	<b>Engenharia de Software</b>	<b>49</b>
3.1.1	Git	49
3.1.2	Github	49
3.1.3	Trello	50
<b>3.2</b>	<b>Elaboração da Monografia</b>	<b>50</b>
3.2.1	OverLeaf	50
3.2.2	Zotero	52
<b>3.3</b>	<b>Ferramentas de Desenvolvimento de Projetos Ágeis</b>	<b>52</b>
3.3.1	Sistemas Operacionais	53

<b>3.4</b>	<b>Ferramentas derivadas da Camada de Requisitos</b>	<b>53</b>
3.4.1	Ferramentas de Elicitação	53
3.4.1.1	<i>Wireframes</i>	54
3.4.1.2	<i>Mockups</i>	54
3.4.1.3	<i>Rich Picture</i>	55
3.4.2	Ferramentas de Modelagem	56
3.4.2.1	Diagrama de Sequência	56
3.4.3	Ferramentas de Análise	57
3.4.4	Ferramentas de Gerência de Requisitos	59
<b>3.5</b>	<b>Desenho da Abordagem</b>	<b>60</b>
<b>3.6</b>	<b>Resumo do Capítulo</b>	<b>60</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>61</b>
<b>4.1</b>	<b>Classificação da Pesquisa</b>	<b>61</b>
4.1.1	Abordagem da Pesquisa	61
4.1.2	Natureza da Pesquisa	62
4.1.3	Objetivos da Pesquisa	62
4.1.4	Procedimentos	62
4.1.4.1	Estudo de Caso	63
<b>4.2</b>	<b>Condução do TCC</b>	<b>64</b>
4.2.1	TCC1	64
4.2.2	TCC2	67
<b>4.3</b>	<b>Cronogramas</b>	<b>69</b>
4.3.1	Cronograma TCC1	69
4.3.2	Cronograma TCC2	69
<b>4.4</b>	<b>Resumo do Capítulo</b>	<b>70</b>
<b>5</b>	<b>REQUIREMENTS SPRINT</b>	<b>71</b>
5.0.1	Questionamentos	71
5.0.2	Desenho da Abordagem <i>Requirements Sprint</i>	72
5.0.3	Princípios para Aplicação da Abordagem	72
<b>5.1</b>	<b>Composição da Equipe</b>	<b>74</b>
<b>5.2</b>	<b>Estruturação das Atividades</b>	<b>75</b>
5.2.1	Apresentação do <i>Workflow</i> da Abordagem	76
5.2.2	Descrição da Programação	77
<b>5.3</b>	<b>Resumo do Capítulo</b>	<b>86</b>
<b>6</b>	<b>ESTUDO DE CASO</b>	<b>87</b>



<b>6.1</b>	<b>Planejamento</b>	<b>87</b>
<b>6.2</b>	<b>Seleção do Caso</b>	<b>88</b>
<b>6.3</b>	<b>Preparar para a Coleta e a Análise dos Dados</b>	<b>88</b>
6.3.1	Composição da Equipe	89
6.3.2	Contextualização Prévia da Abordagem para Priorização de Artefatos	89
<b>6.4</b>	<b>Coleta e Análise de Dados</b>	<b>90</b>
6.4.1	Execução da Programação	91
6.4.1.1	Primeiro Ciclo	91
6.4.2	Segundo Ciclo	95
<b>6.5</b>	<b>Reportar Resultados</b>	<b>99</b>
6.5.1	Clareza da Abordagem	100
6.5.2	Abrangência da Abordagem	100
6.5.3	Flexibilidade da abordagem	101
6.5.4	Agilidade da Abordagem	101
6.5.5	Posicionamento dos Envolvidos sobre a Abordagem	102
<b>6.6</b>	<b>Resumo do Capítulo</b>	<b>108</b>
<b>7</b>	<b>CONCLUSÃO</b>	<b>109</b>
<b>7.1</b>	<b>Considerações Finais</b>	<b>109</b>
<b>7.2</b>	<b>Objetivos Realizados e Questão Respondida</b>	<b>110</b>
<b>7.3</b>	<b>Trabalhos Futuros</b>	<b>111</b>
	<b>REFERÊNCIAS</b>	<b>113</b>
	<b>APÊNDICES</b>	<b>117</b>
	<b>APÊNDICE A – REVISÃO SISTEMÁTICA</b>	<b>119</b>
<b>A.1</b>	<b>Planejamento da Revisão:</b>	<b>119</b>
A.1.1	Desenvolvimento de um Protocolo de Revisão.	119
A.1.1.1	Questões de Pesquisa	119
A.1.1.2	Definição das Bases de Pesquisa	120
A.1.1.3	Palavras-Chave	120
A.1.1.4	Procedimento de Seleção de Artigo	120
A.1.1.5	Avaliação da Qualidade	121
A.1.1.6	Processo de Análise de Artigo	121
A.1.1.7	Processo de Inclusão e Exclusão	121

	<b>APÊNDICE B – ABORDAGEM PRELIMINAR</b>	<b>123</b>
<b>B.1</b>	<b>Proposta Preliminar</b>	<b>123</b>
B.1.1	Esboço da Abordagem Preliminar	124
B.1.2	Questionamentos	125
B.1.3	Desenho da Abordagem Preliminar	125
B.1.4	Princípios para Aplicação da Abordagem	126
<b>B.2</b>	<b>Composição da Equipe</b>	<b>127</b>
<b>B.3</b>	<b>Estruturação das Atividades</b>	<b>128</b>
B.3.1	Descrição da Programação	129

## **ANEXOS** **139**

<b>ANEXO A – AVALIAÇÃO DO PROCESSO DE GESTÃO DE RE- QUISITOS UTILIZANDO UM QUESTIONÁRIO DE DUAS FASES</b>	<b>141</b>
---	------------

# 1 Introdução

O capítulo compreende a contextualização, na qual se procura ambientar o leitor nos tópicos chave do trabalho sendo, principalmente: [Engenharia de Requisitos e Projetos Ágeis](#). A [questão de pesquisa](#), a qual norteará essa pesquisa bem como se pretende responder ao final desse trabalho com maior propriedade e clareza, também é tratada nesse capítulo. Seguem ainda: [justificativa](#), visando acordar as principais contribuições do trabalho; e [objetivos gerais e específicos](#), aderentes à proposta, e já direcionando o leitor para os resultados esperados com a realização desse trabalho. Por fim, tem-se a [organização](#) dessa monografia, distribuída em capítulos.

## 1.1 Contextualização

De acordo com ([TABASSAM SEHRISH E AL-QAHTANE, 2019](#)), qualquer abordagem pode ser usada com qualquer técnica de modelagem, e isso é puramente decidido por engenheiros de requisitos com a ajuda de várias restrições, tais como: tipo de projeto, seu domínio, escopo, ambiente de negócios, usuários finais-alvo e partes interessadas. Tais restrições podem ser entendidas como fatores decisivos para escolha da técnica de modelagem e, conseqüentemente, da abordagem de Engenharia de Requisitos adotada e condizente com essa escolha.

Considerando especificamente o escopo das *startups* de software, tem-se que a elicitação de requisitos é particularmente desafiadora nesse contexto, com base nos estudos realizados pelos autores ([RAFIQ USMAN E BAJWA, 2017](#)). Os autores relatam que esse desafio é intensificado, uma vez que as *startups* atuam em projetos de alta incerteza, os quais são vistos como oportunidades de negócio únicas, bem como de aplicação de tecnologias de ponta. Sendo assim, o domínio, na maioria das vezes, é pouco ou nada conhecido. Esse nicho de empresas ainda carece de outras iniciativas, capazes de mitigar as incertezas. Uma carência evidenciada pelos autores ([RAFIQ USMAN E BAJWA, 2017](#)) é a falta de estudos que investigam como aproximar a camada de Requisitos de Software e a necessidade de desenvolvimento ágil, coerente com entregas rápidas e em atendimento às demandas de mercado.

Com base em ([JUN; QIUZHEN; LIN, 2010](#)), tem-se que a falta de compreensão dos requisitos de um projeto é outro fator que dificulta a realização das atividades compreendidas na Engenharia de Requisitos, pois por vezes os *stakeholders* não possuem pleno

conhecimento de suas necessidades no início do projeto. Assim, mudanças de escopo são comuns ao longo do ciclo de vida do projeto, aumentando os esforços, os custos e os prazos.

Diante do exposto, na sociedade comercial moderna, à medida que a tecnologia da informação se desenvolve bem como dados os variados tipos de negócios e as constantes mudanças ambientais, alinhar os requisitos torna-se cada vez mais intangível, trazendo desafios à Engenharia de Requisitos Tradicional (JUN; QIUZHEN; LIN, 2010).

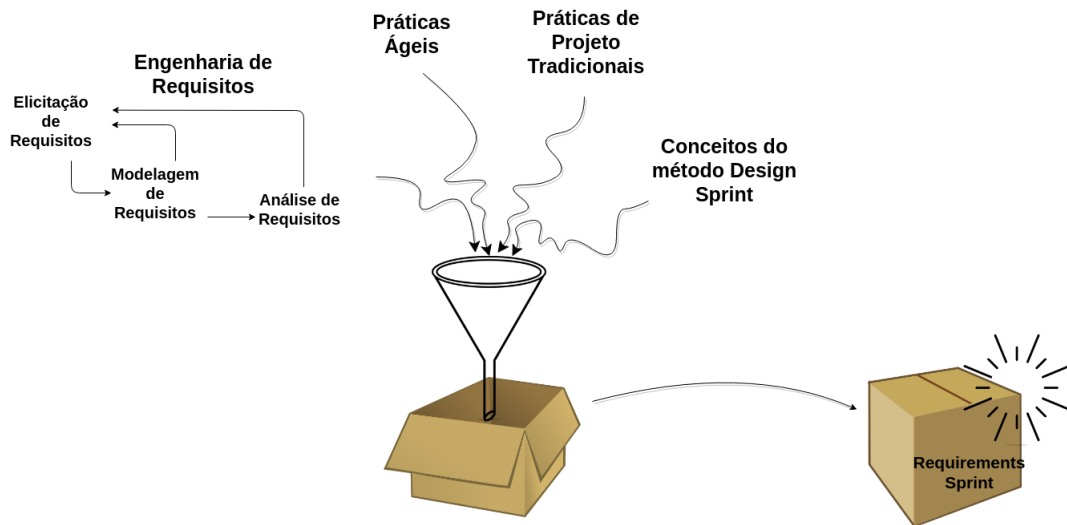
Considerando as dificuldades encontradas na Engenharia de Requisitos Tradicional, surge uma nova abordagem, a Engenharia de Requisitos Ágil (JUN; QIUZHEN; LIN, 2010). Essa procura alterar o status atual da Engenharia de Requisitos Tradicional, buscando conformidade com as constantes mudanças dos requisitos ao longo do ciclo de vida do software. Trata-se de uma proposta de inovação em relação aos métodos tradicionais, sendo a abordagem ágil composta por um conjunto otimizado de práticas ágeis (JUN; QIUZHEN; LIN, 2010).

Enquanto a aplicação de princípios ágeis leva a um melhor sucesso do projeto, alguns projetos ainda falham devido à compreensão insuficiente dos requisitos exatos do cliente. Dessa forma, equipes ágeis começaram recentemente a adotar as práticas do *Design Thinking* para entender melhor o que está na mente dos clientes (PRASAD et al., 2018).

Com o intuito de auxiliar os métodos e as práticas ágeis, a Google Ventures (GV), parte da Google focada em testar e acelerar ideias que ainda estão em estágio inicial de desenvolvimento, lança o método *Design Sprint*, baseado no processo de *design 'express'*. Um processo formado por um grupo de pessoas especialista nas áreas de interesse, que se reúnem durante cinco dias para alinhar, levantar, definir, prototipar e testar ideias, permitindo assim que não seja necessário lançar um Produto Mínimo Viável para descobrir se a ideia é boa ou não, processo esse que pode tomar vários meses. O *Design Sprint* usa princípios do *Design Thinking*, e foca especificamente na validação da ideia com usuários, encurtando o processo para 40 horas de trabalho, conforme explicado por (KNAPP; ZERATSKY; KOWITZ, 2016). Sendo assim, o método *Design Sprint* aproxima as equipes ágeis das práticas de projeto tradicionais (ex. elaboração de protótipo de alta fidelidade e coleta de pontos de vista diferentes).

Tomando como base essa relação harmoniosa entre equipes ágeis e práticas de projeto tradicionais, origina a principal ideia desse Trabalho de Conclusão de Curso. O intuito é aproximar equipes ágeis das práticas da Engenharia de Requisitos, sejam elas tradicionais ou mais emergentes. O método proposto é chamado de *Requirements Sprint*, e foca nas atividades de elicitação, modelagem e análise, buscando inserir conceitos do método *Design Sprint*, conforme apresentado na Figura 1.

Figura 1 – Contextualização da abordagem



Fonte: Autor

## 1.2 Questão de Pesquisa

Tendo como objetivo guiar o trabalho, foi definida a seguinte questão de pesquisa: De que maneira a implementação conjunta de conceitos provenientes das práticas ágeis e do método *Design Sprint* pode contribuir para o desenvolvimento de uma abordagem capaz de apoiar a elicitação, a modelagem e a análise de requisitos em empresas e/ou organizações de pequeno ou médio portes?

## 1.3 Justificativa

Ao longo das décadas, segundo (SPOLETINI; FERRARI, 2017), a importância da elicitação de requisitos tem sido amplamente reconhecida. De fato, essa fase inicial do processo de engenharia de requisitos é relevante para reunir as informações e os dados necessários para especificar os requisitos. Os erros nessa fase podem ser transferidos para as fases subsequentes do desenvolvimento do software e comprometer o processo geral ou aumentar o custo do desenvolvimento (SPOLETINI; FERRARI, 2017).

Com a finalidade de secundar a questão de pesquisa, bem como de se ter uma iteração dedicada às práticas da Engenharia de Requisitos, esse trabalho contribui com a *Requirements Sprint*, sendo essa uma abordagem objetiva e minimalista, focada em elicitação, modelagem e análise de requisitos, mas comprometida com os princípios ágeis,

evitando onerar a equipe com práticas que não agregam valor.

A Figura 2 ilustra um Diagrama de Causa Efeito, visando apresentar as principais causas que dificultam o processo de elicitação, modelagem e análise de requisitos. As principais causas relacionadas são:

- **Gerenciamento:** É muito comum que a organização não tenha um plano de ação ou processo bem definido para execução das atividades do processo de Engenharia de Requisitos;
- **Custo:** Nem sempre a etapa de requisitos é inserida no plano de custos da organização, logo não é investido recurso nesta etapa;
- **Processo:** A maior parte de organizações/empresas/*startups* lidam com um prazo de entrega muito curto para comprovar a viabilidade do produto, demandando assim agilidade no processo inicial de desenvolvimento, e
- **Pessoas:** Nem toda a equipe possui consciência da importância da execução correta deste processo e, por vezes, nem todos os membros possuem experiência para desenvolvimento do mesmo.

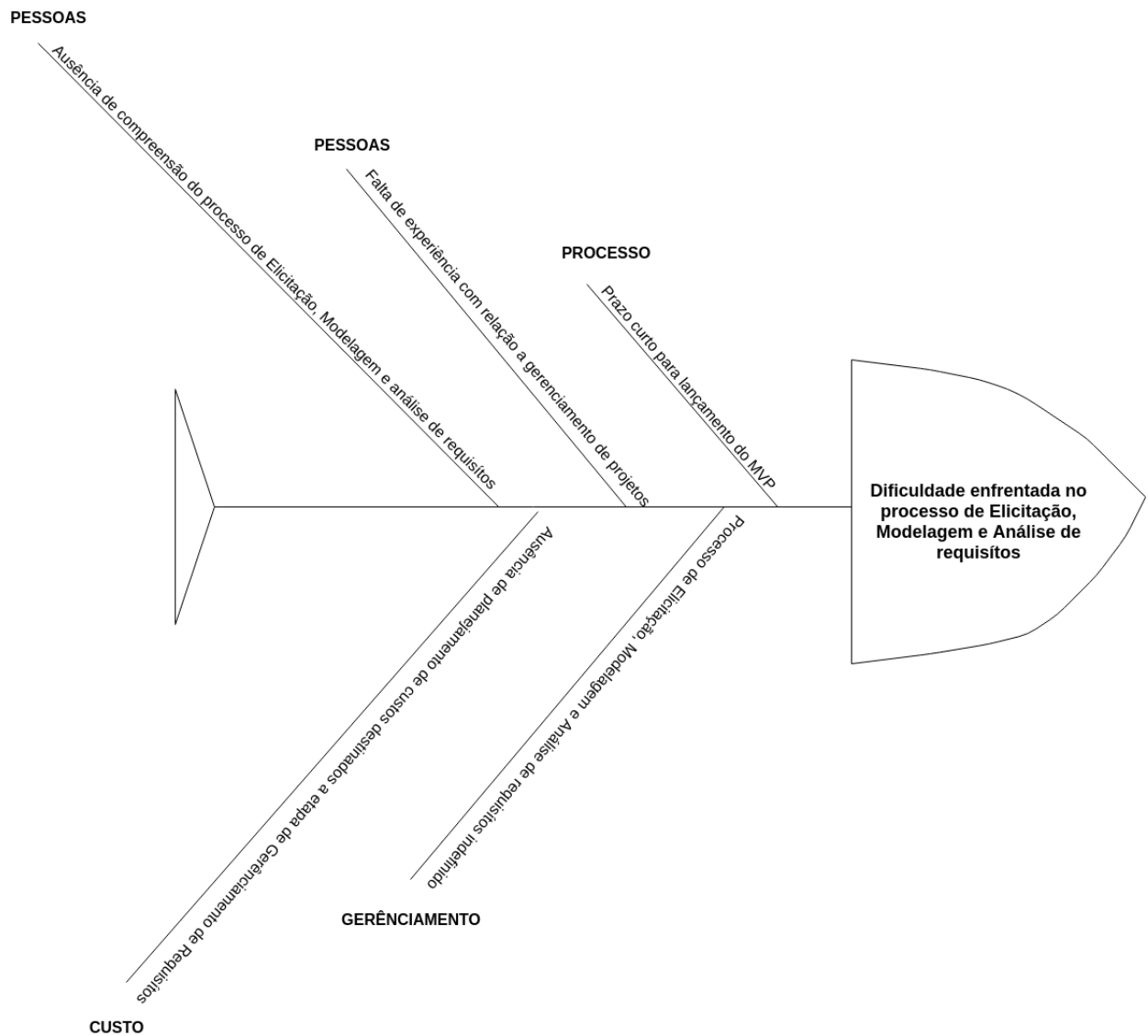
Procurou-se mitigar algumas dessas causas com a elaboração desse trabalho, acordando uma abordagem que não deixa de lado práticas da Engenharia de Requisitos, mas que procura conciliá-las com práticas das equipes ágeis.

### 1.3.1 Revisão Sistemática

Com objetivo de reunir materiais da literatura especializada nos tópicos de interesse desse trabalho, interpretar dados, expandir conhecimentos a serem abordados no referencial teórico, e orientar investigações futuras, foi realizado um estudo em paralelo, baseado em um protocolo de revisão sistemática (KITCHENHAM, 2004), bem como na técnica "*Snow Ball*" especificada em (BALDIN; MUNHOZ, 2011). Detalhes desse estudo encontram-se no Apêndice A.

## 1.4 Objetivo Geral

Elaborar uma abordagem ágil e minimalista, chamada *Requirements Sprint*, focada em elicitação, modelagem e análise de requisitos, que auxilie equipes de projetos ágeis de pequeno e médio portes, as quais encontram-se em estágio inicial de amadurecimento.

Figura 2 – Diagrama de causa e efeito (*fishbone*).

Fonte: Autor

### 1.4.1 Objetivos Específicos

Para realização deste trabalho, foram considerados os seguintes objetivos específicos:

1. Definir, por meio de revisão sistemática, aspectos do método *Design Sprint* a serem utilizados no desenvolvimento da abordagem;
2. Definir, por meio de revisão sistemática, aspectos do Ágil a serem utilizados no desenvolvimento da abordagem;

3. Definir um conjunto de atividades, cujo embasamento é a literatura, de forma que a sua aplicação seja viável de ser realizada em uma *sprint* (*Requirements Sprint*), sendo essa dedicada às práticas da Engenharia de Requisitos e centrada em elicitação, modelagem e análise;

4. Aplicar a abordagem, chamada *Requirements Sprint*, em um projeto ágil. Um segundo plano, caso o estudo de casos não fosse viável, seria o uso de pesquisa ação (RAUPP; BEUREN, 2006), sendo a aplicação da abordagem em um projeto, com as especificações e o escopo elaborados pela autora desse trabalho, e

5. Avaliar os resultados coletados a partir da aplicação do processo resultante da abordagem.

## 1.5 Organização do Trabalho

Este documento está dividido em capítulos, sendo eles:

**Capítulo 2 - Referencial Teórico:** apresenta as áreas e os conceitos relacionados ao tema trabalhado;

**Capítulo 3 - Suporte Tecnológico:** caracteriza as ferramentas, *frameworks* e outros suportes utilizados;

**Capítulo 4 - Metodologia:** especifica as escolhas metodológicas, as principais atividades realizadas ao longo do trabalho, e o cronograma;

**Capítulo 5 - Abordagem Preliminar:** apresenta o conjunto de atividades que compõe a *Requirements Sprint*;

**Capítulo 6 - Status Atual do Trabalho:** apresenta os principais resultados.



## 2 Referencial Teórico

Dado que o desenvolvimento da abordagem atua em dois escopos: Engenharia de Requisitos e Projetos Ágeis, neste capítulo, será apresentado o referencial teórico associado aos dois tópicos. Com esse intuito, faz-se necessário sintetizar os conceitos utilizados, com a finalidade de acordar um breve levantamento quanto aos referenciais que norteiam essa pesquisa. Dentre os conceitos abordados, têm-se destaque aos relacionados às atividades de [elicitacão](#), [modelagem](#) e [análise](#) no escopo da Engenharia de Requisitos, bem como aos relacionados às [práticas ágeis aplicáveis em projetos de pequeno e médio portes](#). Comenta-se ainda sobre o método *Design Sprint*, fonte de inspiração nesse trabalho, visando aproximar equipes ágeis às práticas tradicionais da Engenharia de Software, sem prejuízo aos aspectos considerados relevantes para essas equipes, tais como: agilidade no desenvolvimento, sendo esse iterativo e incremental, bem como documentação suficiente e comunicação entre os envolvidos. Por fim, tem-se o [resumo do capítulo](#)

### 2.1 Considerações Iniciais

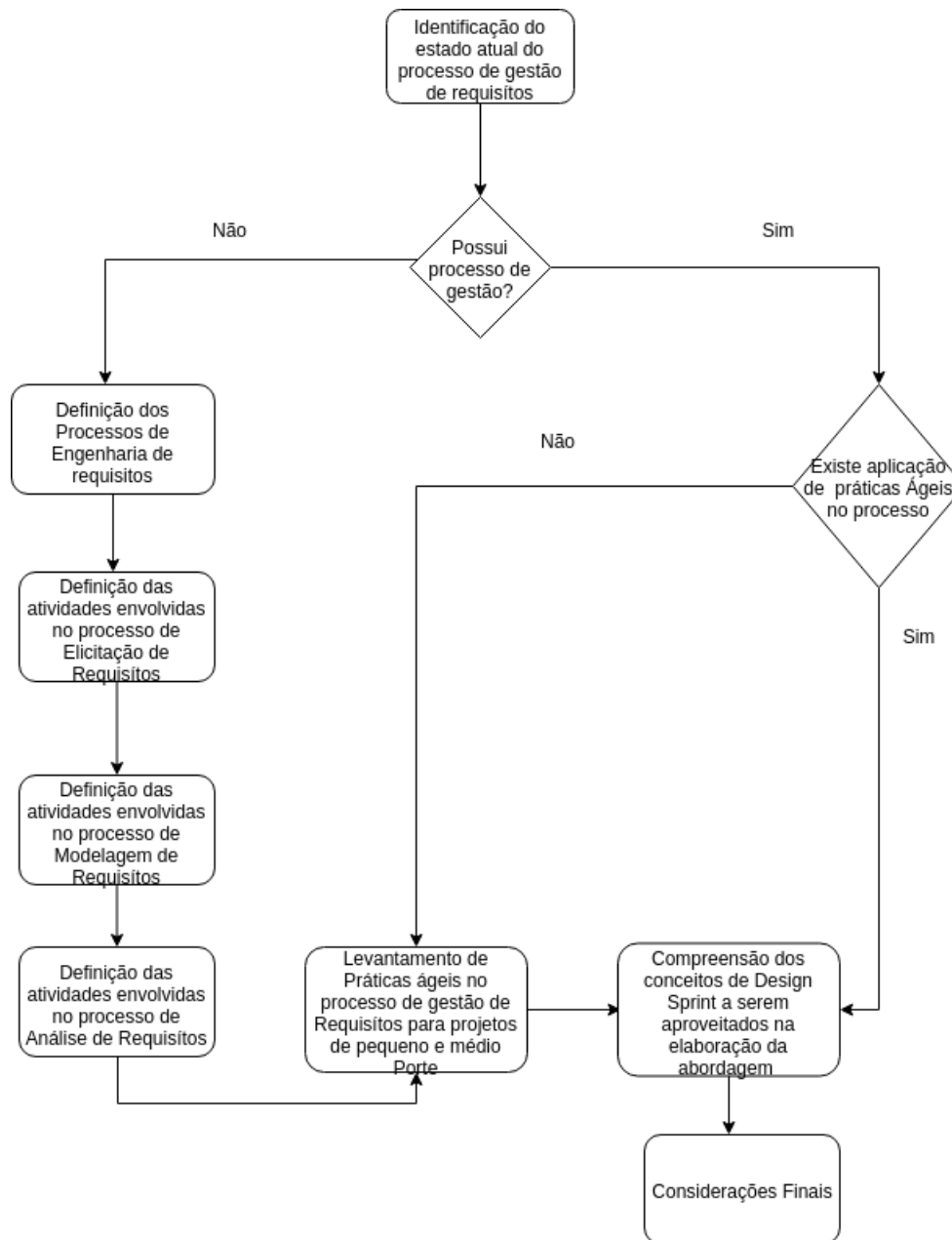
Visando situar o leitor no que tange aos tópicos mais relevantes para essa pesquisa, foi elaborado um diagrama de processo de leitura, Figura 3. Esse modelo especifica um fluxo simples e retroalimentado, o que confere uma cadência aos tópicos cobertos nesse capítulo, interligando-os para maior e melhor compreensão por parte do leitor. Outros detalhes sobre cada tópico serão tratados ao longo do capítulo, nas seções 2.2 a 2.5.

### 2.2 Engenharia de Requisitos

De acordo com ([SOMMERVILLE; SAWYER, 1997](#)), a Engenharia de Requisitos é o processo de descoberta, documentação e gerenciamento dos requisitos para um sistema baseado em computador. O objetivo da Engenharia de Requisitos é produzir um conjunto de requisitos do sistema que, na medida do possível, seja completo, consistente, relevante e reflita o que o cliente realmente deseja. Embora esse ideal seja provavelmente inatingível, o uso de uma abordagem sistemática baseada em princípios de engenharia tende a levar a melhores requisitos do que a abordagem informal que ainda é comumente usada.

Conforme descrito por ([RIAZ et al., 2018](#)), existem muitos fatores que podem desempenhar um papel na taxa de sucesso de projetos para qualquer organização, mas téc-

Figura 3 – Diagrama de processo de leitura.



Fonte: Autor

nicas de Engenharia de Requisitos utilizadas pela organização é o fator-chave. De acordo com o estudo (RIAZ et al., 2018), falhas durante as fases da Engenharia de Requisitos podem influenciar significativamente os processos de desenvolvimento de qualquer projeto. Riaz et al. (2018) apontam ainda que os requisitos não descobertos durante as fases de Engenharia de Requisito podem custar 200 vezes mais para a correção nos estágios posteriores.

Os autores ([ARIAS; BUCCELLA; CECHICH, 2018](#)) enfatizam que informações estruturadas e organizadas podem ser exploradas com menos esforço, para encontrar componentes reutilizáveis quando novos requisitos aparecem. A fim de apoiar este processo, a gestão de requisitos surge, de acordo com ([CUEVAS; SERRANO; SERRANO, 2004](#)), como uma abordagem sistemática para identificar, documentar, organizar e rastrear todos os requisitos. Nesse contexto, o principal objetivo é estabelecer um acordo entre o cliente e a equipe de software sobre o significado dos requisitos. A qualidade na gestão de requisitos é fundamental para o sucesso do processo de software.

Segundo ([ÁVILA; SPÍNOLA, 2007](#)), os requisitos de software podem ser classificados em:

- Requisitos funcionais: são requisitos diretamente ligados à funcionalidade do software, ou seja, descrevem as funções que o software deve executar;
- Requisitos não funcionais: são requisitos que expressam condições que o software deve atender ou qualidades específicas que o software deve ter. Em vez de informar o que o sistema fará, os requisitos não-funcionais colocam restrições no sistema, e
- Requisitos de domínio: são requisitos derivados do domínio da aplicação e descrevem características do sistema e qualidades que refletem o domínio. Podem ser requisitos funcionais novos, restrições sobre requisitos existentes ou computações específicas.

### 2.2.1 Identificação do Estado Atual do Processo de Gestão de Requisitos

Os autores em ([CUEVAS; SERRANO; SERRANO, 2004](#)) defendem a ideia de que, embora um processo de gestão de requisitos não seja realizado em muitas organizações, há pessoas dentro das organizações que executam algumas práticas de gestão de requisitos. No entanto, essas práticas geralmente não são documentadas e como consequência não são disseminadas pela organização.

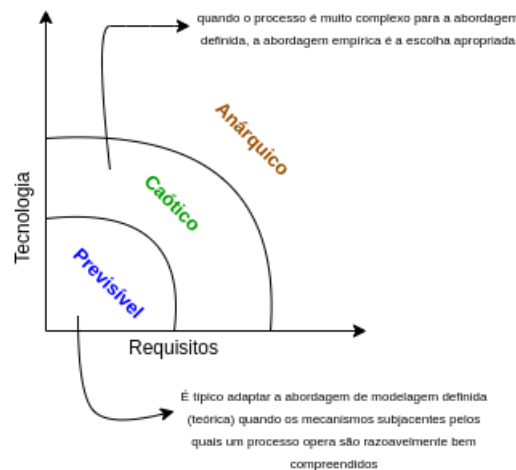
Foi realizada uma pesquisa baseada em um Estudo de Casos, pelos autores ([CUEVAS; SERRANO; SERRANO, 2004](#)), na qual foi proposta uma metodologia de avaliação do estado atual do processo de gestão de requisitos amparado pelo uso de um questionário de dois estágios, a fim de se identificar quais práticas do processo de gestão de requisitos são executadas, mas não documentadas pela organização; quais práticas exigem ser priorizadas, e quais não são implementadas devido à má gestão ou desconhecimento.

Ressalta-se ainda que o questionário proposto pelos autores baseou-se nas práticas dos processos de gestão de requisitos estabelecidos pelo CMMI (Modelo Integrado de Maturidade em Capacitação).Entretanto, não houve viabilidade de aplicação como fonte

de embasamento para identificação do estado atual da organização. Em um primeiro momento, a intenção era identificar o nível de maturidade da organização, mas, não houve retorno por parte dos autores (CUEVAS; SERRANO; SERRANO, 2004), mesmo diante da solicitação via canais de comunicação divulgados pelos autores nos materiais de pesquisa consultados. Portanto, foi registrado no Anexo A os resultados alcançados, visando manter os rastros para maior embasamento da abordagem aqui estabelecida.

Apesar da impossibilidade de aplicação do questionário para identificação do nível de maturidade, essa carência não inviabiliza ou mesmo gera prejuízo significativo à presente pesquisa, uma vez que será assumido que, se a organização aplica a metodologia Scrum, ela segue os preceitos dessa metodologia, sendo, portanto, classificada como *Chaotic*, conforme ilustrado na Figura 4

Figura 4 – Classificação da metodologia segundo *Scrum*.



Fonte: (JAMES; WALTER, 2010-2018)

Segundo (JAMES; WALTER, 2010-2018), o Scrum é um *framework* empírico, sendo apropriado para trabalhar com requisitos incertos e/ou tecnologias incertas, mas não desconhecidas por completo pela organização. Dessa forma, o Scrum atende bem às organizações classificadas como *Chaotic*, e não *Anarchy*; tendo portanto certo nível de maturidade no desenvolvimento naquele domínio. Adicionalmente, não é muito aderente às organizações classificadas como *Predictable*, uma vez que essas, por terem um conhecimento bastante razoável do domínio, podem utilizar abordagens até mesmo bem tradicionais, sem prejuízo ao desenvolvimento.

Em resumo, tem-se que será assumido como estado atual do processo de gestão de requisitos para as organizações candidatas ao uso da presente abordagem: organizações *Chaotic*, com práticas de gestão de requisitos incipientes, mas que conhecem - mesmo que

de forma limitada - o domínio para o qual pretendem desenvolver uma solução computacional.

## 2.3 Atividades dos Processos de Engenharia de Requisitos

### 2.3.1 Atividade de Elicitação de Requisitos

Conforme referido por (RAFIQ USMAN E BAJWA, 2017), a elicitação de requisitos é considerada a primeira, principal e relevante atividade dos processos de Engenharia de Requisitos. Inclui atividades que pretendem descobrir, adquirir e elaborar requisitos para sistemas de software.

Rafiq Usman e Bajwa colocam que as informações recolhidas durante a elicitação de requisitos, muitas vezes tem de ser interpretadas, analisadas, modeladas e validadas antes para que os engenheiros se sintam confiantes de que um conjunto "completo" de requisitos de um sistema seja recolhido. Adicionalmente, argumentam que a seleção de técnicas de elicitação de requisitos tem dependência com o problema, a solução, o domínio e os requisitos existentes.

Wohlin et al. (2005) descrevem cinco tarefas típicas na elicitação de requisitos:

- Entendimento do domínio da aplicação: exame do mundo real em que o sistema será inserido, os objetivos e problemas que demandam o sistema, analisando aspectos políticos, sociais e organizacionais, bem como as restrições para o desenvolvimento do sistema;

- Identificação das fontes de requisitos: registro das principais fontes de coleta de requisitos, dentre elas a identificação dos *stakeholders*, notadamente os usuários do sistema, e dos outros sistemas que interagem com o sistema de interesse, bem como a seleção da documentação existente relacionada com o sistema de interesse;

- Análise dos *stakeholders*: análise e envolvimento dos *stakeholders* na Engenharia de Requisitos;

- Seleção de recursos: escolha de técnicas, abordagens e ferramentas a serem utilizadas na elicitação, e

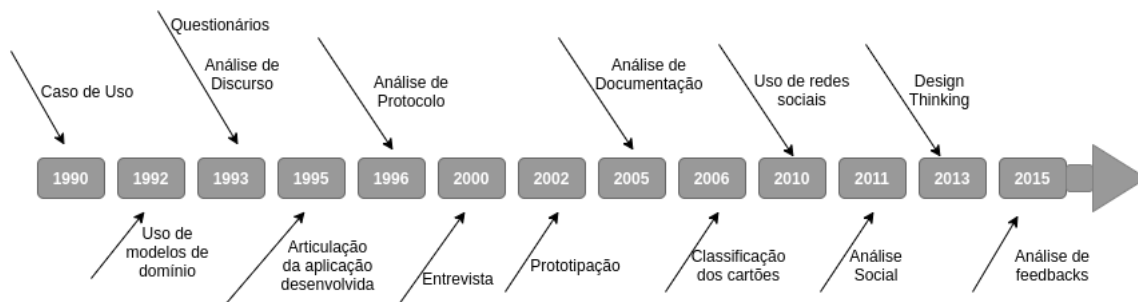
- Elicitação dos requisitos propriamente dita: definição da abrangência e dos limites do sistema, investigação das necessidades dos *stakeholders*, análise do ambiente onde o sistema será inserido e como ele endereçará os principais objetivos e problemas levantados, a fim de ajudar o analista na descrição de um conjunto de requisitos essenciais.

De acordo com (RAFIQ USMAN E BAJWA, 2017), a prática de elicitação de re-

quisitos depende de vários fatores, incluindo o tipo de sistema, tipo de desenvolvimento de software e grau de adaptabilidade. Foi argumentado que a maioria das pequenas empresas é encontrada desenvolvendo sistemas que têm requisitos orientados para o mercado. Esta característica pode também ser encontrada em *startups* de software, embora sejam entidades diferentes do que as pequenas empresas que possuem uma única missão. No caso das *startups*, há anseio por um escalonamento rápido, apesar de seu tamanho pequeno inicialmente.

Na Figura 5, é apresentado um *timeline* referente às técnicas de elicitação de requisitos. Entrevistas constam como as principais técnicas em 1990. O surgimento e o uso de muitas técnicas podem ser destacados entre os anos de 1993 a 2002. A partir de 2006, há menção à reutilização de requisitos, análise social, *design thinking* e análise de *feedback*, deixando clara uma tendência de alinhamento entre ser mais ágil, reaproveitando conhecimento elicitado, bem como uma preocupação com critérios de qualidade e pontos de vista diferentes.

Figura 5 – *Timeline* referente às técnicas de elicitação de requisitos.



Fonte: (RAFIQ USMAN E BAJWA, 2017)

Os autores (RAFIQ USMAN E BAJWA, 2017) especificam que a elicitação de requisitos se torna mais complicada quando a natureza dos requisitos é orientada para o mercado. Requisitos tradicionais utilizam-se de técnicas de elicitação, nas quais as fontes de elicitação são clientes ou usuários, não sendo apropriadas para levantar requisitos de mercado por exemplo. Em requisitos orientados para o mercado, não há clientes ou usuários específicos até que seja liberada pelo menos a versão beta para um grupo de controle que representa uma amostra do público alvo. Nesse cenário, requisitos orientados pelo mercado frequentemente acabam sendo "inventados" em um primeiro momento.

Em uma análise mais profunda, os autores continuam suas colocações, relatando que, com base no estudo realizado nas empresas, foi identificado que o processo de "invenção" é inspirado em estratégias de negócios e na visão de um produto. Posteriormente,

são validados e complementados com a análise de mercado. Portanto, os requisitos geralmente não são bem especificados, e a linguagem natural é considerada suficiente para comunicar os requisitos. Em algumas situações, os requisitos são ainda comunicados verbalmente. As principais fontes de elicitação de requisitos em organização de mercados de software são desenvolvedores, pessoal de vendas e gestores. Em alguns casos, os desenvolvedores são encontrados entre os usuários pretendidos de um produto. Portanto, eles "inventam" requisitos baseados nas suas compreensão e imaginação.

Uma vez o produto liberado, a elicitação é feita examinando relatórios de *bugs*, *feedback* direto do usuário e por meio de técnicas de serviços de apoio. No entanto, a partir desse ponto, e segundo (RAFIQ USMAN E BAJWA, 2017), quando os requisitos parecem ficar mais conhecidos, há uma escassez de estudos revelando como a elicitação dos requisitos é conduzida em *startups*. Tal escassez causa a sensação da não existência de registros, ou mesmo do não gerenciamento desses requisitos até a implantação ou a disponibilização do software. Portanto, os autores foram além, e conduziram um estudo para investigar como as *startups* de software atuam nesse sentido.

Três casos foram selecionados após uma estratégia de amostragem estratificada e prática, abrangendo diferentes tipos de aplicativos (*Mobile vs. Web-based*) e estágios de desenvolvimento (ideação, funcional ou maduro). São referidos como Startup 1, 2 e 3, respectivamente, para a confidencialidade. Conforme apresentado na Tabela 1 a seguir.

Tabela 1 – Tabela referente ao estudo de casos sobre técnicas de elicitação de requisitos em *startups*.

<b>Perfil</b>	<b>Startup 1</b>	<b>Startup 2</b>	<b>Startup 3</b>
Localização	Canadá	Itália	Brasil
Lançada em	2014	2014	2012
N de Fundadores	2	2	5
N de Funcionários	2	6	25
N de Desenvolvedores	1	1	5
Estágio do Produto	Conceitual	Funcional	Maduro
Domínio	Saúde	Comercial	Comercial
Tipo de Software	Aplicação <i>Mobile</i>	Aplicação <i>Mobile</i> e <i>Web</i>	Aplicação <i>Web</i>

Fonte: (RAFIQ USMAN E BAJWA, 2017) Tradução realizada pelo Autor

Segundo (RAFIQ USMAN E BAJWA, 2017), a maior coleta de dados foi realizada por meio de entrevistas com perguntas abertas. O projeto das perguntas da entrevista foi guiado pelos requisitos revistos pelas técnicas de elicitação apresentadas na Figura 5. Para cada inicialização, o diretor de tecnologia ou o desenvolvedor sênior foi entrevistado como o informante chave. Cada entrevista durou entre 60 minutos a 90 minutos. Todas as entrevistas foram gravadas e transcritas de forma textual para análise. Além disso, os *sites* das *startups* foram inspecionados cuidadosamente para obter um melhor entendimento de suas ofertas de produtos e outras informações.

Os autores descrevem ainda o processo, especificando que, antes do início do mesmo, os fundadores das *startups* já tinham algo em mente sobre o que construir. Com o passar do tempo, os requisitos são refinados e evoluídos praticando um número de atividades com um foco principal na pesquisa do usuário. A pesquisa do usuário é conduzida aplicando diversas técnicas de elicitação de requisitos. Este é o caso das *Startups* 1 e 3. Por exemplo, a ideia inicial da Startup 3 foi um clone de alguns negócios existentes em outro País. Mas a ideia era ainda validada, embora informalmente, através de táticas de pesquisa do usuário antes de qualquer tipo de desenvolvimento de produto. Em vez disso, a *Startup* 2 não suscitava requisitos das pessoas inicialmente. A ideia inicial foi fornecida por um dos cofundadores, e os primeiros requisitos foram "inventados" pela equipe. A Tabela 2 relaciona as técnicas utilizadas nas *startups* estudadas. Entre essas técnicas, entrevistas e prototipagem têm sido comumente utilizadas, conforme apresentado.

O estudo destaca que três técnicas adicionais foram encontradas sendo utilizadas apenas na *Startup* 1, entre elas: questionários, uso de rede social e *feedback* por meio da análise de comentários. Estas três técnicas foram de alguma forma entrelaçadas e aplicadas em conjunto. Além de aplicar o envio de *e-mail* para identificar o que as pessoas precisam, outras iniciativas merecem menção, sendo: consulta aos fóruns de usuários relevantes e mídia social; leitura de *posts* e perguntas, e observação das reações dos usuários.

Os resultados indicam que os processos de elicitação de requisitos em *startups* são primordiais e, principalmente, informais, sendo contínuos e caminhando em paralelo à evolução do produto. A abordagem geral de elicitação em *startups* é centrada numa ideia pré-concebida. Não há clientes identificados ou partes interessadas para extrair os requisitos diretamente. Conseqüentemente, as soluções concebidas baseiam-se nos pressupostos e na compreensão dos fundadores do mercado.

Em (RAFIQ USMAN E BAJWA, 2017), menciona-se ainda que os achados deste estudo destacam diversas técnicas de elicitação de requisitos utilizadas em *startups* de software, incluindo técnicas relatadas na literatura como: prototipagem, entrevistas, debate informal, questionários, utilização de redes e análise de comentários de *feedback*. Há



Tabela 2 – Tabela de especificação da utilização de técnicas de elicitação por *startups*.

<b>Técnicas de Elicitação de Requisitos</b>	<b>Startup 1</b>	<b>Startup 2</b>	<b>Startup 3</b>
TÉCNICAS EXISTENTES			
Entrevista	Sim	Sim	Sim
Prototipação	Sim	Sim	Sim
<i>Brainstorming</i>	Sim	Sim	Não
Questionários	Sim	Não	Não
Utilizando <i>Network Social</i>	Sim	Não	Não
Análise de <i>Feedbacks</i> por meio de comentários	Sim	Não	Não
NOVAS TÉCNICAS			
Análise de produtos similares ou concorrentes	Sim	Sim	Sim
Invenção da equipe colaborativa	Sim	Sim	Sim
Envolvimento de modelos de usuário	Sim	Sim	Não

Fonte: (RAFIQ USMAN E BAJWA, 2017) Tradução realizada pelo Autor

menção ainda às técnicas: análise de produtos similares ou concorrentes, e discussões de equipe colaborativa.

Cabe ressaltar que a análise de concorrentes, equipe colaborativa de discussão, prototipagem e entrevistas são técnicas comuns entre as três *startups* em diferentes estágios do produto. Além disso, comparando a lista abrangente de técnicas de elicitação relatadas na literatura, Tabela 2, parece que as *startups* usam poucas técnicas recomendadas na literatura. Como mostrado anteriormente, a *Startup 1* (produto no estágio conceitual) empregou todas as técnicas. Entretanto, quando o produto se torna mais concreto, menos técnicas de elicitação estão envolvidas, como mostrado nas *Startups 2* e 3. Isto pode ser tomado como indicação de que a intensidade e a frequência de utilização técnicas são diferentes durante diferentes estágios do produto.

Por fim, os autores concluem que o estudo mostra que as *startups* não especificam ou definem requisitos explicitamente. Portanto, nenhuma documentação formal foi preparada durante ou após o processo de elicitação de requisitos. As únicas documentações encontradas foram na forma de notas, listas de tarefas e *e-mails* de *feedback*. Uma possível explicação reside na "Cultura *Startup*", que é diferente da cultura corporativa tradicional

que se orienta por um processo de elicitação.

### 2.3.2 Atividade de Modelagem de Requisitos

Em (ÁVILA; SPÍNOLA, 2007), os autores especificam que uma vez identificados e negociados, os requisitos devem ser documentados para que possam servir de base para o restante do processo de desenvolvimento. Entre os muitos problemas que enfrentamos na documentação de requisitos, certamente, administrar o grande volume de informações gerado pelo processo de requisitos é um dos principais.

De acordo com (TURINE; MASIERO, 1996), a atividade de modelagem tem por objetivo criar e desenvolver modelos que descrevem estática e dinamicamente o que o sistema deve fazer, e não como deve ser feito. Estes modelos expressam os requisitos descritos no documento de requisitos, possibilitando um maior entendimento do domínio da aplicação, servindo para determinar se a especificação está completa, consistente e precisa, e fornecendo uma transição para a fase de projeto.

Assim, Turine e Masiero especificam que devido à importância do documento de requisitos dentro do processo de desenvolvimento do software, é fundamental que este documento seja organizado de forma a melhorar a compreensão e a legibilidade dos requisitos, evitando que problemas e erros surjam na fase de implementação do software.

Logo, em (ÁVILA; SPÍNOLA, 2007), os autores reiteram a importância dos requisitos serem documentados em um nível apropriado de detalhe. Em geral, é produzido um documento de especificação de requisitos, de forma que todos os *stakeholders* possam entendê-lo. O registro dos requisitos em um documento próprio facilita o controle de alterações de todos os envolvidos na manutenção dos requisitos, bem como a geração de versões do documento e a facilidade de acesso por todos os envolvidos.

### 2.3.3 Atividade de Análise de Requisitos

Para (TURINE; MASIERO, 1996), a atividade de análise de requisitos é fundamental para o sucesso do processo de desenvolvimento do software. Nesta atividade, o projetista (engenheiro de requisitos) especifica as funções e o desempenho do software, indica a interface para integração com outros sistemas, e estabelece as restrições de projeto do software.

O objetivo da análise de requisitos é avaliar e revisar o escopo do software (documento de requisitos do software). Através de um processo de descoberta, refinamento, modelagem e especificação, segundo (TURINE; MASIERO, 1996), a intenção é obter uma especificação de requisitos "completa" e consistente.

Diante das colocações dos autores, é muito provável que o documento de requisitos obtido até então possua várias inconsistências ou problemas de funcionalidade, e o projetista, durante esta fase, deve ser capaz de detectar e resolver inconsistências. As decisões de análise servem para realimentar e melhorar o documento de requisitos do sistema, pois este será a base para todas as fases da Engenharia de Software subsequentes.

Esta atividade examina, segundo (ÁVILA; SPÍNOLA, 2007), a especificação do software, de forma a assegurar que todos os requisitos foram definidos sem ambiguidades, inconsistências ou omissões, detectando e corrigindo possíveis problemas ainda durante a fase de definição dos requisitos. Neste contexto, sabe-se que o custo da correção de defeitos aumenta na medida em que o processo de desenvolvimento progride.

Revisões de artefatos de software têm se mostrado, também com base nas colocações dos autores em (ÁVILA; SPÍNOLA, 2007), uma abordagem eficiente e de baixo custo para encontrar defeitos logo após terem sido introduzidos, reduzindo o retrabalho e melhorando a qualidade dos produtos. Não é em vão que modelos de maturidade de processo de software, como o CMMI, exigem a condução de revisões.

Um tipo particular de revisão de software, citada por (ÁVILA; SPÍNOLA, 2007), é o uso de inspeções. Inspeções possuem um processo de detecção de defeitos rigoroso e bem definido. Os benefícios de se aplicar inspeções são maiores para artefatos desenvolvidos no início do processo, como o documento de requisitos.

O autor (FAGAN, 2002), responsável pela criação do processo de inspeção, provou ser um processo extremamente eficaz como meio de reduzir o número de defeitos que os usuários encontram no software, bem como aumentar a produtividade do desenvolvimento. Isso foi possível por meio da utilização de inspeções para encontrar defeitos e removê-los de todos os produtos de trabalho que são criados no decurso do desenvolvimento de software, nomeadamente em especificações, projeto, codificação, planos de teste, casos de teste, entre outros.

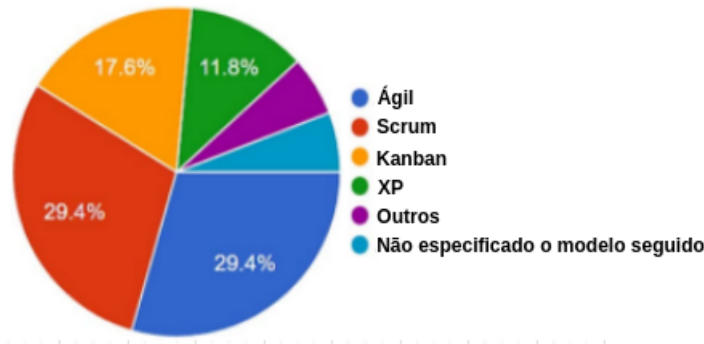
Segundo Fagan, os componentes necessários para fazer inspeções de software de forma bem-sucedida incluem três princípios:

- Definição formal do processo, ou seja, garantir que cada membro da equipe esteja ciente dos objetivos, da função e dos critérios de entrada e de saída de cada fase de processo;
- Processo de inspeção, sendo esse um processo de sete etapas utilizado para encontrar defeitos, e
- Melhoria contínua do processo visando a remoção de defeitos sistêmicos.

## 2.4 Métodos Ágeis

Por meio de um levantamento realizado por (RIAZ et al., 2018) sobre o tipo de modelo seguido pela maioria das organizações, o método Ágil é o mais utilizado, conforme apresentado na Figura 6

Figura 6 – Levantamento do método mais utilizado para Engenharia de Requisitos.



Fonte: (RIAZ et al., 2018)

Segundo (VLAANDEREN et al., 2011), uma das principais inovações no desenvolvimento de metodologias de software dos últimos anos tem sido a introdução de princípios ágeis. Desde a criação do manifesto ágil em 2001, incluindo os anos que anteciparam a sua criação, vários métodos ágeis de desenvolvimento de software entraram em prática. Alguns exemplos são o Scrum (VLAANDEREN et al., 2011), *Extreme Programming* (XP) (AWAD, 2005) e *Feature Driven Development* (FDD) (AWAD, 2005).

Scrum é basicamente uma estrutura ágil e leve que fornece etapas para gerenciar e controlar o processo de desenvolvimento de software e produtos. Scrum é a combinação do modelo iterativo e o modelo incremental, uma vez que as compilações são sucessivas e incrementais em termos de recursos para desenvolver o software. Com base em (SRIVASTAVA; BHARDWAJ; SARASWAT, 2017), trata-se de uma metodologia projetada para aumentar a velocidade de desenvolvimento, alinhar o indivíduo e as organizações, definir uma cultura centrada no desempenho, apoiar o acionista na criação de valor, para ter uma boa comunicação de desempenho em todos os níveis e melhorar o desenvolvimento individual e a qualidade de vida.

Awad, em (AWAD, 2005), descreve o XP como um processo caracterizado por curtos ciclos de desenvolvimento, planejamento incremental, *feedback* contínuo, confiança na comunicação, e no projeto evolutivo. Com todas as qualidades, os programadores respondem ao ambiente em mudança com muito mais coragem.

Já o FDD, segundo (AWAD, 2005), possui um processo metodológico mais curto e que se difere do processo das demais metodologias, pois não abrange todo o desenvolvimento do processo de software, centrando-se nas fase de desenho e execução de funcionalidades para o projeto. Seu processo de desenvolvimento é guiado por um modelo, uma lista de funcionalidades e o planejamento ocorre por funcionalidade, no início do projeto.

Para esse Trabalho de Conclusão de Curso, o Scrum foi escolhido como a metodologia ágil base para o desenvolvimento da abordagem, dada a sua popularidade na comunidade ágil. A utilização de conceitos e práticas das demais metodologias abordadas não é uma impossibilidade, uma vez que cada um dos referidos métodos possuem práticas relevantes para contribuição com a abordagem concebida.

### 2.4.1 Práticas Ágeis do Scrum

Em concordância com (AWAD, 2005), o Scrum não requer ou fornece quaisquer métodos/práticas de desenvolvimento de software específicos para ser usado. Em vez disso, requer certas práticas de gerenciamento e ferramentas em diferentes fases do Scrum para evitar o caos por imprevisibilidade e complexidade.

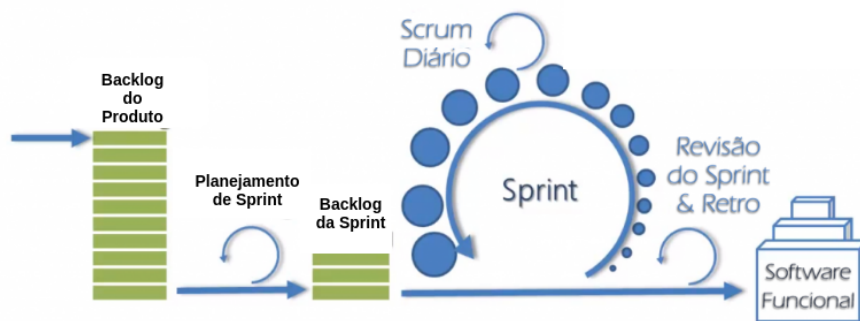
Em (SRIVASTAVA; BHARDWAJ; SARASWAT, 2017), os autores referem-se ao Scrum como um processo que envolve um *Scrum Master*, o *Product Owner* e uma Equipe Scrum. O principal papel do *Scrum Master* é eliminar impedimentos. A Equipe *Scrum* é uma estrutura funcional que compreende programadores, testadores e outros peritos de vários campos exigidos no desenvolvimento que conduz a um produto final em atendimento à satisfação do cliente.

Segundo (AWAD, 2005), *Sprint* é o procedimento de adaptação às variáveis ambientais em mudança como: requisitos, tempo, recursos, conhecimento, tecnologia, dentre outros. Deve ainda resultar em um incremento potencialmente lançável de software. As ferramentas de trabalho da equipe são: reuniões de planejamento de *Sprint*, *Sprint backlog* e Reuniões diárias do Scrum. A duração de uma *sprint* pode variar de uma a três semanas, como especificado em (SRIVASTAVA; BHARDWAJ; SARASWAT, 2017).

Com isso, algumas práticas ágeis são definidas por (SRIVASTAVA; BHARDWAJ; SARASWAT, 2017) como parte do processo metodológico do Scrum. Dentre essas práticas, têm-se: desenvolvimento da *Sprint backlog*, uma documentação de todos os requisitos para *Sprint* atual a ser trabalhada. A *Sprint Backlog* é uma lista de requisitos que são determinados pelo *Product Owner*, resultantes do processo de Engenharia de Requisitos que são chamadas de histórias de usuário. Ela é seguida pela prática de planejamento de *Sprint*, que inclui métodos para se obter a conclusão da mesma. No final de cada dia,

é realizada a prática do *Scrum* diário que visa o progresso da tarefa atribuída para o dia. O objetivo da *Sprint* é entregar um produto potencialmente lançável. No final de cada *Sprint*, ocorre a prática de revisão da *Sprint*, na qual são apresentadas as atividades realizadas ao proprietário do produto para demonstrar o lançável incremento, conforme apresentado na Figura 7.

Figura 7 – Processo do *framework Scrum*.



Fonte: (SCRUM, 2019)

## 2.4.2 Projetos Ágeis de Pequeno e Médio Portes

De acordo com (ZHANG; DORN, 2012), o desenvolvimento ágil tem recebido interesse crescente, tanto na indústria quanto na Academia, devido aos seus benefícios no desenvolvimento de software rapidamente, atendendo às necessidades dos clientes, e mantendo o ritmo.

O desenvolvimento ágil visa à satisfação do cliente por meio do fornecimento de componentes de software úteis, desenvolvidos por um processo iterativo, com foco em requisitos mínimos. Usar métodos ágeis ajuda a refinar a viabilidade e apoiar o processo para obter *feedback* rápido à medida que uma funcionalidade é introduzida.

Métodos ágeis têm sido sugeridos como forma de maximizar o valor do negócio através de equipes pequenas e autoorganizáveis, usando tecnologias flexíveis e envolvimento precoce dos clientes para evolução constante do software, especialmente em pequenos projetos (LEE; YONG, 2013). Para (ZHANG; DORN, 2012), a equipe é autoorganizável, quando o líder da equipe não decide qual pessoa vai fazer qual tarefa ou como um problema será resolvido. O que ocorre de fato é que a equipe como um todo toma tal decisão. A equipe é *cross-functional* de modo que todos são necessários para realizar uma tarefa da ideia à execução.

Lee e Yong (2013) acordam que métodos ágeis são altamente atraentes para pequenos projetos, mas nenhum método ágil funciona bem como um sistema autônomo. Portanto, alguma adaptação ou personalização é sempre necessária. Pequenas e/ou médias empresas representam até 85% de todas as empresas de software em Países que a indústria de TI é avançada. Geralmente, essas empresas, segundo (LEE; YONG, 2013), são muito responsivas e flexíveis, e estão diretamente/indiretamente envolvidas em muitos pequenos projetos. Os desafios na gestão de pequenos projetos vêm do planejamento e da capacidade de resposta precisando ser alinhados em pouco tempo. Por outro lado, os desafios na gestão de projetos maiores consistem em analisar a complexidade das áreas de negócio, tecnologias, riscos e gerenciar um número maior de *stakeholders*. Assim, pequenos e médios projetos exigem uma abordagem diferente do que projetos maiores.

Há alguns fatores que Lee e Yong (2013) determinam para que seja possível classificar um projeto de pequeno ou médio porte. Um pequeno ou médio projeto geralmente:

- Dura seis meses ou menos;
- Pode envolver trabalho a tempo;
- Tem dez ou menos membros na equipe;
- Envolve um pequeno número de áreas de habilidade;
- Tem um único objetivo e solução que é prontamente alcançável;
- Tem um escopo e uma definição estritamente definidos;
- Afeta uma única unidade de negócio e tem um único tomador de decisões;
- Não requer soluções automatizadas de fontes de projeto externas;
- Não tem implicações políticas;
- Produz entregáveis diretos sem interdependências entre as áreas de habilidade, e
- Tem financiamento disponível.

Dada esta definição, os mesmos autores descreveram ainda cinco desafios para pequenos e médios projetos que utilizam abordagens ineficazes, tais como:

- Falta de planejamento;
- Baixa prioridade;
- Equipes de projetos inexperientes;
- Gerente de projeto responsável por múltiplas funções, e
- Utilização de ferramentas e processos de gestão de projetos para pequenos pro-

jetos.

Apesar do grande número de pequenos projetos em todo o mundo, segundo (LEE; YONG, 2013), há uma pequena quantidade de literatura discutindo soluções de Engenharia de Software que seja focada em apenas pequenos projetos. Para (LEE; YONG, 2013), métodos ágeis são uma alternativa leve aos métodos tradicionais, e baseiam-se em quatro grandes processos: desenvolvimento iterativo, *feedback* do cliente, pequenas equipes de desenvolvimento de software e tecnologias flexíveis de software.

Lee e Yong (2013) destacam ainda que pequenos projetos não são apenas versões reduzidas de grandes e não são necessariamente mais fáceis para gerenciar do que grandes projetos. Eles ainda podem ser complexos e envolver uma variedade de departamentos, recursos, fornecedores e clientes para completá-los. O desenvolvimento de software ágil teve um enorme impacto sobre como o software é desenvolvido em todo o mundo. No entanto, embora existam muitos métodos ágeis, que são conhecidos por serem eficazes para pequenos e médios projetos, pouco se sabe sobre soluções práticas de adoção e como os resultados são avaliados.

Ágil é um conceito guarda-chuva, segundo (LEE; YONG, 2013), abrangendo métodos diferentes e cada método usando seu próprio vocabulário e sua própria terminologia. Como os projetos diferem em sua escala, escopo e desafios técnicos, um método não se adequa a todas as circunstâncias. Assim, os profissionais de software possuem a necessidade de selecionar e aplicar cuidadosamente as práticas ágeis mais benéficas de acordo com os requisitos do projeto.

### 2.4.3 *Design Sprint* - Método Base para a Abordagem Desenvolvida

De acordo com a Google Ventures, a *Design Sprint* é um processo de cinco dias para responder perguntas críticas de negócios através de *design*, prototipagem e testes de ideias com os clientes. É um "*Greatest Hits*" da estratégia de negócios, inovação, ciência do comportamento, *Design Thinking*, entre outros, embalado em um processo de trabalho intenso.

Na Google Ventures, as *Sprints* são versáteis e executadas em empresas como Nest, Flatiron Health e Medium — para ajudá-los a entrar em novos mercados, projetar novos produtos, desenvolver novos recursos para milhões de usuários, definir estratégias de *marketing* e outros anseios. Logo, é especificada a necessidade de se definir, antes de se iniciar a *Sprint*, o desafio e a equipe correta. Faz-se necessário ainda dispor de recursos como tempo e espaço para conduzir a *Sprint*. O método *Design Sprint* é baseado em cinco dias de desenvolvimento, conforme listado a seguir. Será feito o uso de primeira pessoa,



uma vez que a escrita que se segue orienta-se pelo próprio linguajar usado pela *Google Venture* aos seus interessados. Trata-se de um linguajar mais informal, e que atende mais diretamente ao público interessado em consumir as práticas do método *Design Sprint*.

1. Segunda-Feira: as discussões estruturadas de segunda-feira criam um trajeto para a semana da *Sprint*. De manhã, você começará e no final concordará com um objetivo de longo prazo. Em seguida, você vai fazer um mapa do desafio. À tarde, você pedirá aos especialistas da sua empresa que compartilhem o que eles sabem. Finalmente, você escolherá um alvo: uma parte ambiciosa mas gerenciável do problema que você pode resolver em uma semana;

2. Terça-Feira: depois de um dia cheio de compreensão do problema e escolhendo um alvo para sua *Sprint*, na terça-feira, você começa a se concentrar em soluções. O dia começa com inspiração: uma revisão das ideias existentes para remixagem e aperfeiçoamento. Então, à tarde, cada pessoa vai esboçar, seguindo um processo de quatro etapas que enfatiza o pensamento crítico sobre a arte. Você também começará a planejar o teste de cliente da sexta-feira recrutando clientes que se encaixam no seu perfil de destino;

3. Quarta-Feira: na quarta-feira de manhã, você e sua equipe terão uma pilha de soluções. Isso é ótimo, mas também é um problema. Você não pode prototipar e testá-los todos — você precisa de um plano sólido. Na parte da manhã, você vai criticar cada solução, e decidir quais têm a melhor chance de alcançar o seu objetivo a longo prazo. Então, à tarde, você vai levar as cenas vencedoras de seus esboços e criar um *storyboard*: um plano passo-a-passo para o seu protótipo;

4. Quinta-Feira: na quarta-feira, você e sua equipe criaram um *storyboard*. Na quinta-feira, você adotará uma filosofia "falsa" para transformar esse *storyboard* em um protótipo. Uma fachada realista é tudo que você precisa para testar com os clientes, e aqui está a melhor parte: concentrando-se na superfície voltada para o cliente do seu produto ou serviço, você pode terminar o seu protótipo em apenas um dia. Na quinta-feira, você também vai se certificar de que tudo está pronto para o teste de sexta-feira, confirmando o cronograma, revendo o protótipo, e escrevendo um roteiro de entrevista, e

5. Sexta-Feira: sua *Sprint* começou com um grande desafio, uma equipe excelente e não muito mais. Até sexta-feira, você criou soluções promissoras, escolheu a melhor, e construiu um protótipo realista. Isso só seria possível em uma semana impressionantemente produtiva. Mas, você vai dar um passo adiante, entrevistar os clientes e aprender assistindo-os reagir ao seu protótipo. Este teste faz toda a *Sprint* valer a pena: no final do dia, você vai saber o quão longe você tem que ir, e o que fazer em seguida.

## 2.5 Resumo do Capítulo

Neste capítulo, foram abordados conceitos introdutórios sobre a Engenharia de Requisitos bem como sua impotência com relação à identificação do estado atual do processo de gestão da organização, no qual foi acordado que será assumido de que a organização possui um nível de maturidade suficiente, baseado nos princípios ágeis, tornando possível a integração da abordagem desenvolvida à linhagem de desenvolvimento da organização.

Em seguida, foram explicitadas as atividades existentes no processo de desenvolvimento de requisitos: elicitacão, modelagem e análise. Essas fases estão presentes como atividades imprescindíveis na abordagem desenvolvida, com a finalidade de se ter, ao final da *Requirements Sprint*, uma *baseline* de requisitos o mais clara e concisa possível. Logo, foi acordado também que será empregada a metodologia Scrum como guia principal para a inclusão de práticas ágeis na *Requirements Sprint*. Entretanto, essa decisão não exclui a possibilidade de inclusão de alguma prática referente à outra metodologia citada.

Associado à inserção de práticas ágeis na abordagem desenvolvida, encontram-se presentes conceitos do método *Design Sprint*. Pretende-se que esses conceitos permitam - uma vez adaptados para o escopo da Engenharia de Requisitos - que a *baseline* de requisitos seja deliberada em uma *sprint*, já com bom escopo de especificação, e com tendência de facilitar a implantação da abordagem proposta em projetos de pequeno e médio portes.

## 3 Suporte Tecnológico

Neste capítulo, são apresentadas as tecnologias que foram utilizadas no desenvolvimento deste trabalho de monografia. Conforme apresentado, a seção de [Engenharia de Software](#) confere uma visão sobre as ferramentas utilizadas com relação à gestão do trabalho. Na seção seguinte, é apresentado o contexto referente à [Elaboração da Monografia](#), no qual são apresentadas as ferramentas relacionadas ao apoio de pesquisa e escrita do trabalho. É abordado também, neste, capítulo uma seção referente ao [Desenho da Abordagem](#), no qual se descreve a ferramenta que foi utilizada para o desenho da mesma. Por fim, tem-se a seção referente ao [Resumo do Capítulo](#).

### 3.1 Engenharia de Software

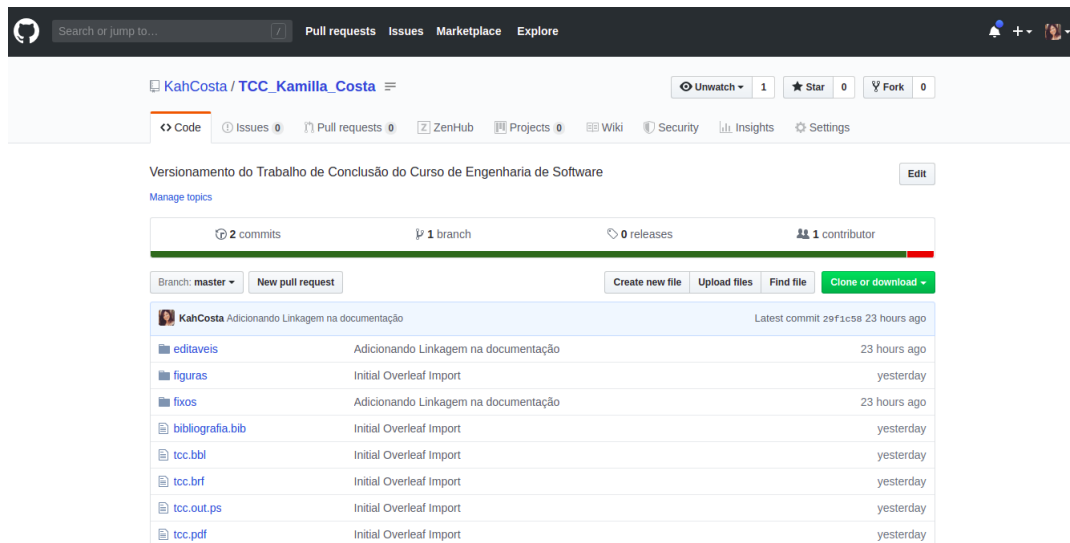
As ferramentas mencionadas a seguir, da seção [3.2.1](#) à seção [3.2.4](#), foram utilizadas com a finalidade de apoio à gestão do trabalho, de modo a apresentar as metodologias e ferramentas utilizadas no desenvolvimento do trabalho.

#### 3.1.1 Git

O Git consiste em um sistema de controle de versão distribuído, gratuito e *open-source*, que permite o versionamento de documentos de arquivo, códigos ou imagens. Com a finalidade de se obter uma boa gestão, foi decidido utilizar o Git ([GIT, 2019](#)) para armazenamento e versionamento tanto da monografia, quanto dos artefatos produzidos ao longo desse trabalho.

#### 3.1.2 Github

Plataforma de armazenamento de arquivos para controle de versão e colaboração, permitindo assim que várias pessoas trabalhem simultaneamente em um mesmo projeto. O Github ([GITHUB, 2019](#)) possui alguns atributos atrativos como: a possibilidade de criação de repositórios públicos ou privados, revisão de código, documentação e funcionalidades de integração que facilitam o seu uso por comunidades. A Figura 8 ilustra o uso dessa plataforma para controle de versão da monografia.

Figura 8 – Utilização da plataforma *github* na monografia.

Fonte: Autor

### 3.1.3 Trello

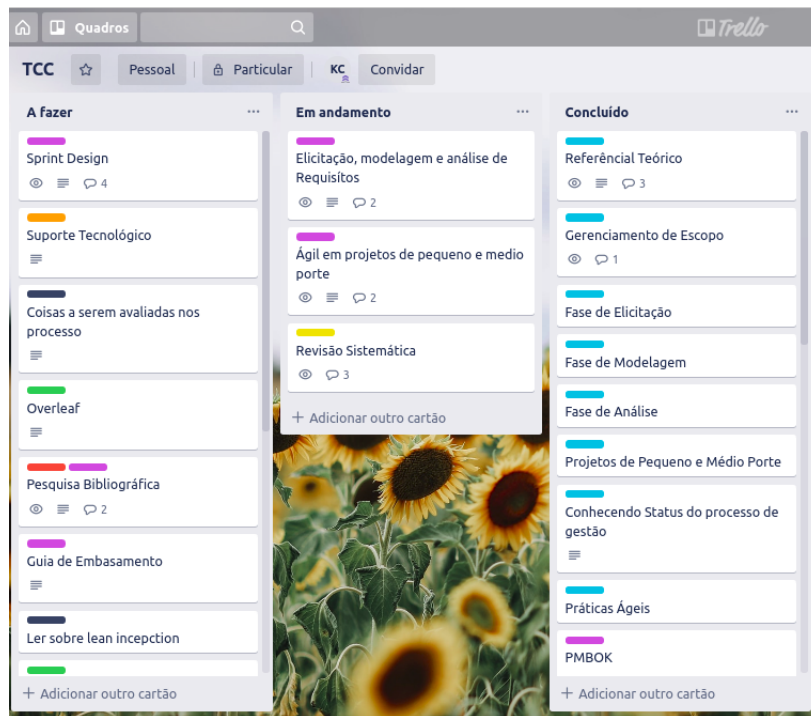
O Trello ([TRELLO, 2019](#)) é um aplicativo de gerenciamento de projetos baseado na web originalmente, que opera um modelo de negócio. É utilizado para gerenciar atividades e tópicos de estudo, bem como a escrita da monografia. A Figura 9 mostra o quadro de acompanhamento do presente projeto, sendo esse gerenciado no Trello e organizado em listas "A fazer", "Em andamento" e "Concluído", uma espécie de Kanban. Além disso, são utilizados vários cartões, cada qual referenciando um tópico de pesquisa, tal como: "Pesquisa Bibliográfica" e "Referencial Teórico". Tal prática permite um acompanhamento passo a passo do projeto em andamento.

## 3.2 Elaboração da Monografia

Esta seção compreende a exposição das ferramentas que foram utilizadas para a escrita da monografia. A mesma encontra-se organizada em subseções, conforme apresentado a seguir.

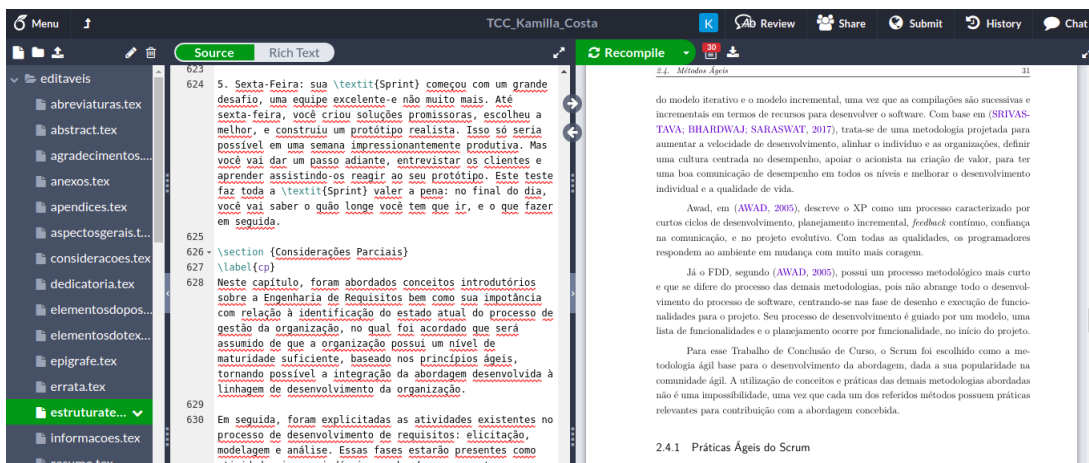
### 3.2.1 OverLeaf

Baseia-se em um editor e um compilador *online* de LaTeX (Sistema de preparação de documentos com formatação de alta qualidade). Esse suporte é aplicado com frequência para documentos técnicos ou científicos, de médio ou grande porte. O OverLeaf ([OVER-](#)

Figura 9 – Utilização da ferramenta *trello* para gerenciamento do projeto.

Fonte: Autor

LEAF, 2019) pode ser utilizado para quase todos os tipos de publicações. A Figura 10 ilustra o uso do OverLeaf na edição da presente monografia.

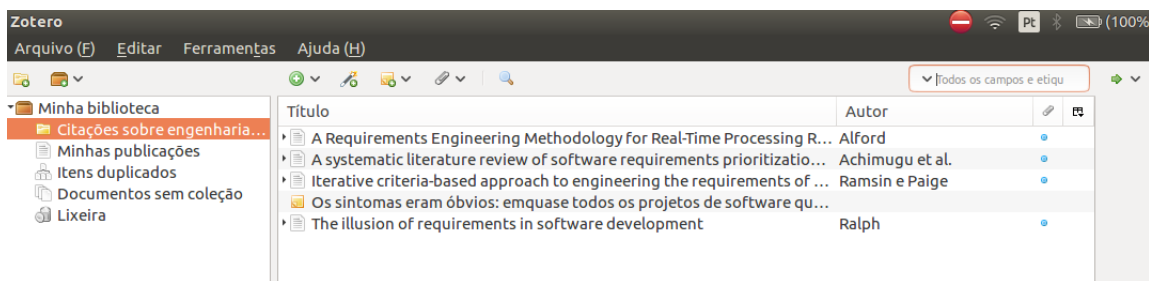
Figura 10 – Utilização do editor *overLeaf* na edição da monografia.

Fonte: Autor

### 3.2.2 Zotero

As referências foram gerenciadas por meio da utilização do Zotero (ZOTERO, 2019), sendo essa uma ferramenta de código aberto, que apoia na coleta, organização, análise e compartilhamento de pesquisas, permitindo o armazenamento de autores, títulos e outros campos da pesquisa e, posteriormente, a exportação de todos os campos preenchidos como referências formatadas. A Figura 11 mostra a utilização da ferramenta Zotero para gerenciamento das referências que embasaram o presente trabalho. Um recurso bastante utilizado foi a possibilidade de distribuir as referências em pastas, conferindo maior clareza e organização ao referencial teórico do projeto.

Figura 11 – Utilização do *zotero* para gerenciamento das referências desse trabalho.



Fonte: Autor

## 3.3 Ferramentas de Desenvolvimento de Projetos Ágeis

As metodologias ágeis de desenvolvimento são usadas, segundo (GAEA, 2019), para acelerar a construção de soluções tecnológicas. Portanto, tem por objetivo a melhoria contínua dos processos, trazendo avanços de comunicação e interação entre a equipe e os usuários, mais organização para o alcance de metas, diminuição de erros e retrabalhos, mais colaboração e, sobretudo, respostas rápidas às mudanças. Tudo isso favorece a geração de mais produtividade para os desenvolvedores, além de redução de custos e até mais satisfação com o trabalho.

Dado que a abordagem proposta foca em Projetos Ágeis de Pequeno e Médio Portes bem como na área de Requisitos de Software, sentiu-se a necessidade de amparar os interessados na abordagem com uma proposta de suporte tecnológico que vai desde um olhar mais focado em sistemas operacionais específicos, até mesmo em ferramentas para apoio às atividades de Elicitação, Modelagem e Análise de Requisitos. Sendo assim, têm-se que as próximas seções acordam esse referencial tecnológico sugerido.

Lembrando que a abordagem tem uma perspectiva flexível, visando maior aderência do público alvo à abordagem. Assim, o referencial tecnológico é apenas sugestivo, e não impositivo, sendo perfeitamente possível a equipe ágil adaptar esse conjunto de tecnologias conforme queiram. A ideia é apenas direcionar, principalmente à equipes iniciantes, e que, portanto, podem não ter ou conhecimento, ou experiência com um arcabouço tecnológico apropriado.

### 3.3.1 Sistemas Operacionais

Projetos de desenvolvimento ágeis de software podem ser realizados por meio de execução em múltiplos sistemas operacionais, sendo os mais comuns: *Windows*, *IOS* e associados ao *Linux*. Neste contexto, com relação à aplicação da abordagem, foi definido que a mesma se concentra em software livre, no qual concede liberdade ao usuário para executar, acessar, modificar o código fonte, e redistribuir cópias com ou sem modificações. Sendo assim, a abordagem está direcionada a oferecer suporte, acordando um referencial tecnológico compatível e mais direcionado aos projetos em sistemas operacionais derivados do *Linux*, como: *Ubuntu*, *Mint*, *Debian* e outros.

## 3.4 Ferramentas derivadas da Camada de Requisitos

Quando aplicativos, *sites*, sistemas, ou diversos outros ambientes de interação são iniciados, o primeiro passo a ser executado é a criação de diferentes tipos de propostas de como será a aplicação e de como se dará seu comportamento. Com isto, foi selecionado um conjunto de ferramentas de apoio para as atividades de Elicitação, Modelagem e Análise inseridas no processo da abordagem.

### 3.4.1 Ferramentas de Elicitação

Para execução da atividade de Elicitação, são propostas algumas ferramentas voltadas para a elaboração de *wireframes* e *Mockups*. A ideia é permitir que sejam elaborados protótipos de baixa, média e/ou alta fidelidades, sendo a prototipação uma técnica de elicitação muito agregadora, e que permite rapidamente ao cliente e à equipe terem um noção mais concreta do que será a solução computacional de fato.

Para embasar melhor o processo de elicitação, sugere-se o uso de *Rich Pictures*, conforme consta na seção 3.4.1.3. Trata-se de uma "figura rica", a qual acorda determinados fluxos chave do domínio em investigação bem como os principais envolvidos e as principais funcionalidades.

### 3.4.1.1 Wireframes

VelhoBit (2019) define *Wireframes* como leiautes demonstrativos da diagramação de elementos na tela, que geralmente possuem indicadores. Trata-se de uma evolução de um esboço, sendo mais bem trabalhado, indicando a disposição dos diversos elementos e conteúdo. Por originalmente serem desenhados em papéis milimetrados ou pautados, são também chamados de esboços de grade. Esta etapa ainda pode ser feita por alguém sem muita experiência com *design*, pois é basicamente uma forma de organizar as ideias rabiscadas anteriormente no papel.

A Figura 12 ilustra a ferramenta sugerida para construção de wireframes: (CACCO, 2019), uma ferramenta *online* de desenho que permite a criação de *wireframes*. Como resultado, espera-se um conjunto de prótótipos de baixa fidelidade, construídos, de preferência, de forma colaborativa entre clientes e equipes ágeis. Esse processo permitirá acordar as principais funcionalidades e não funcionalidades (questões de usabilidade, principalmente), gerando uma espécie de esboço inicial da solução computacional desejada.

Adicionalmente, facilitará elicitare conhecimento tácito, sendo esse um conhecimento que o cliente é detentor, mas não revela facilmente, ou por achar que é muito óbvio, ou por esquecimento, ou por ser algo que para ele é intrínseco ao contexto. Mas, normalmente, não o é para a equipe, o que causa inconsistências e insucessos na elaboração da solução computacional.

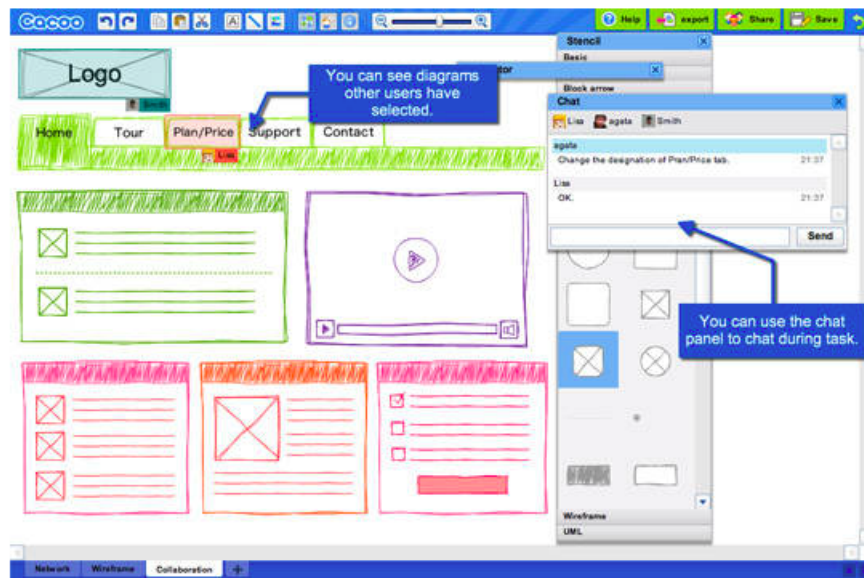
#### Especificações da Ferramenta:

- Plataforma *Web*;
- Exporta *wireframe* em formato de imagem PNG, e
- Edição de forma colaborativa, permitindo que mais de uma pessoa possa trabalhar em cima do *wireframe*.

### 3.4.1.2 Mockups

VelhoBit (2019) define *mockups* como leiautes com exemplos finais, ou muito próximo destes, para maior conhecimento da interface gráfica. Basicamente, serve de mapa para a implementação da mesma, com exemplos das telas, componentes e estilos já finalizados. Os *mockups* são bem mais trabalhados que os *wireframes* e raramente possuem indicadores internos, para não atrapalhar a compreensão da *GUI* (*Interface Gráfica do Usuário*). Muitas vezes, é comum mostrar os *mockups* também em cenas de interações, ajudando a demonstrar os fluxos aos interessados bem como ajustar questões de usabi-



Figura 12 – Ferramenta para elaboração de *wireframes*.

Fonte: (CACCO, 2019)

lidade, tendo como base uma ideia mais concreta quanto ao resultado final da solução computacional. A ideia é elaborar protótipos de média e alta fidelidades.

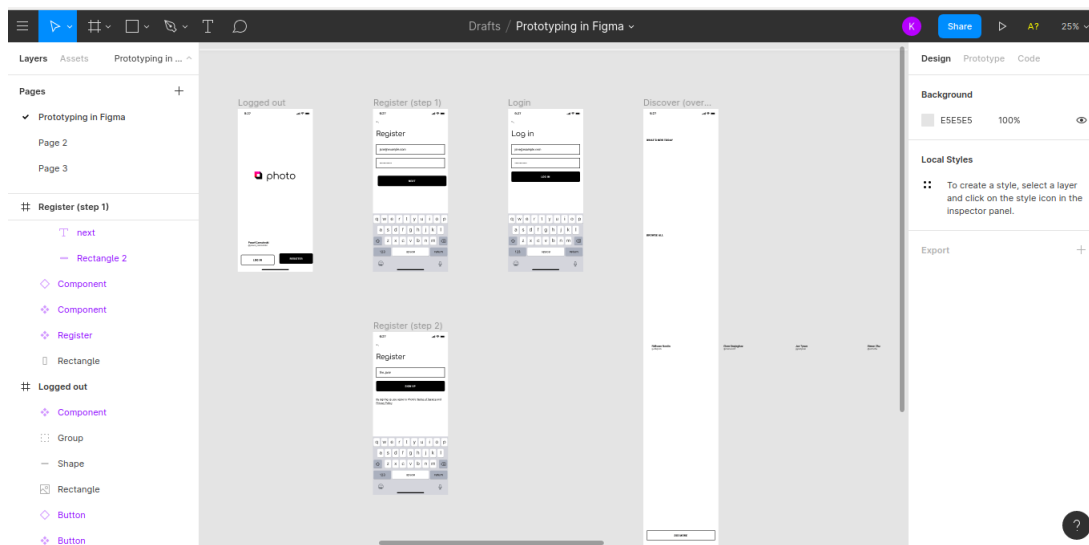
A Figura 13 ilustra a ferramenta sugerida para construção de *Mockups* (FIGMA, 2019):

#### Especificações da Ferramenta:

- Plataforma *Web*;
- Edição de forma colaborativa, permitindo que mais de uma pessoa possa trabalhar em cima do *mockup*, e
- Permite definição de rotas de forma iterativa e simples.

#### 3.4.1.3 *Rich Picture*

Um *Rich Picture*, segundo (BETTEREVALUATION, 2019), é uma maneira de explorar, reconhecer e definir uma situação e expressá-la através de diagramas para criar um modelo mental preliminar. Um *Rich Picture* ajuda a abrir a discussão e chegar a um entendimento amplo e compartilhado de uma situação, conforme ilustrado na Figura 14.

Figura 13 – Ferramenta para elaboração de *mockups*.

Fonte: (FIGMA, 2019)

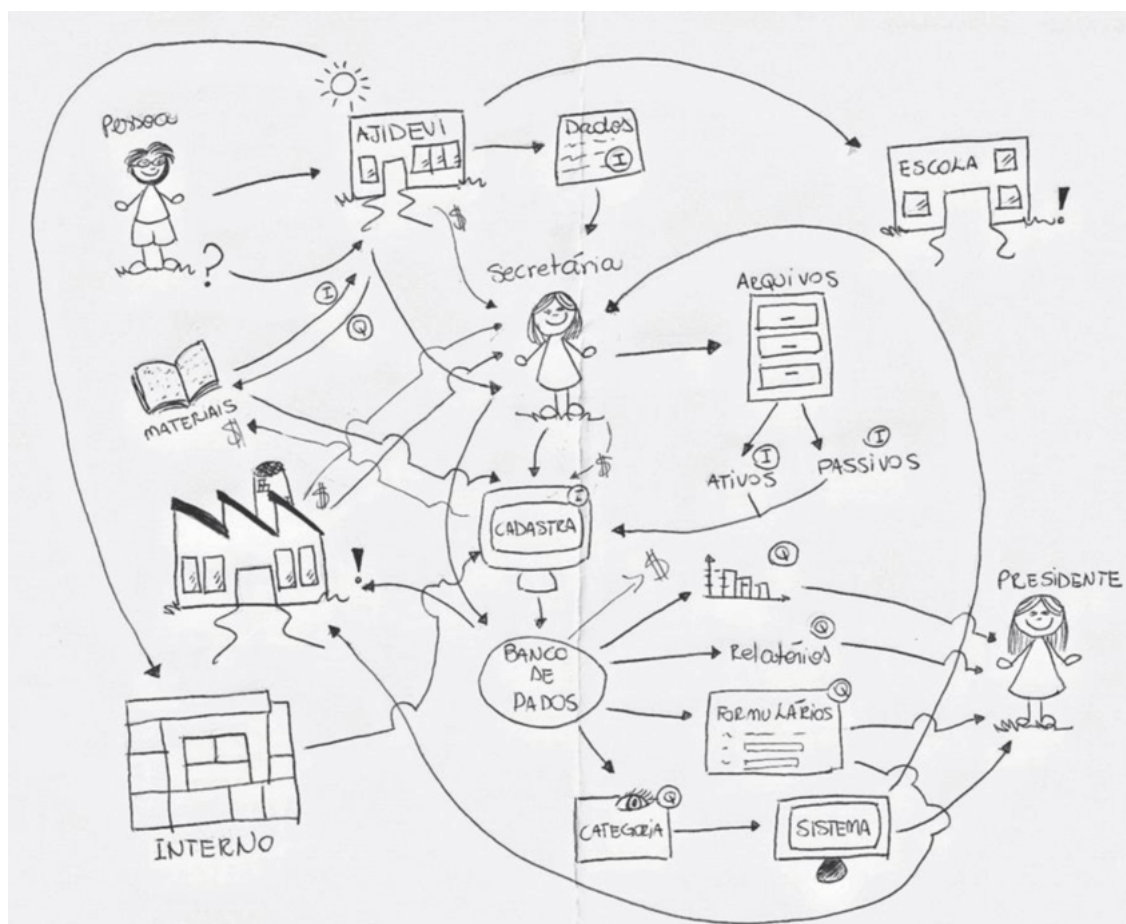
### 3.4.2 Ferramentas de Modelagem

Após os requisitos serem elicitados, é necessário que os mesmos sejam modelados de forma a se tornarem claros em termos de níveis de abstração mais baixos, ou seja, mais próximo das linguagens e notações utilizadas pelas equipes técnicas. Além disso, busca-se auxiliar na elaboração do *Backlog*, de forma que seja possível construir uma visão mais unificada e uniforme do produto como um todo.

Para isso, e com base em autores como o (LARMAN, 2004), que defendem a elaboração de diagramas dinâmicos e notações que conferem uma visão comportamental no desenho de uma solução computacional, sugere-se o uso de um "modelo" na notação da UML, sendo: um típico diagrama dinâmico, chamado Diagrama de Sequência.

#### 3.4.2.1 Diagrama de Sequência

De acordo com (LUCIDCHART, 2019), diagrama de sequência é uma espécie de diagrama de interação, pois descreve como, e em qual ordem, um grupo de objetos trabalha em conjunto. Estes diagramas são usados por desenvolvedores de software e profissionais de negócios para entender as necessidades de um novo sistema ou para documentar um processo existente. A Figura 15 ilustra a ferramenta sugerida para construção de diagrama de sequência: uma ferramenta *online* de desenho que permite a criação de diagramas UML, chamada CACCO (CACCO, 2019), conforme apresentado na Figura 15.

Figura 14 – Exemplo de *Rich Picture*.

Fonte: ([BETTEREVALUATION, 2019](#))

#### Especificações da Ferramenta:

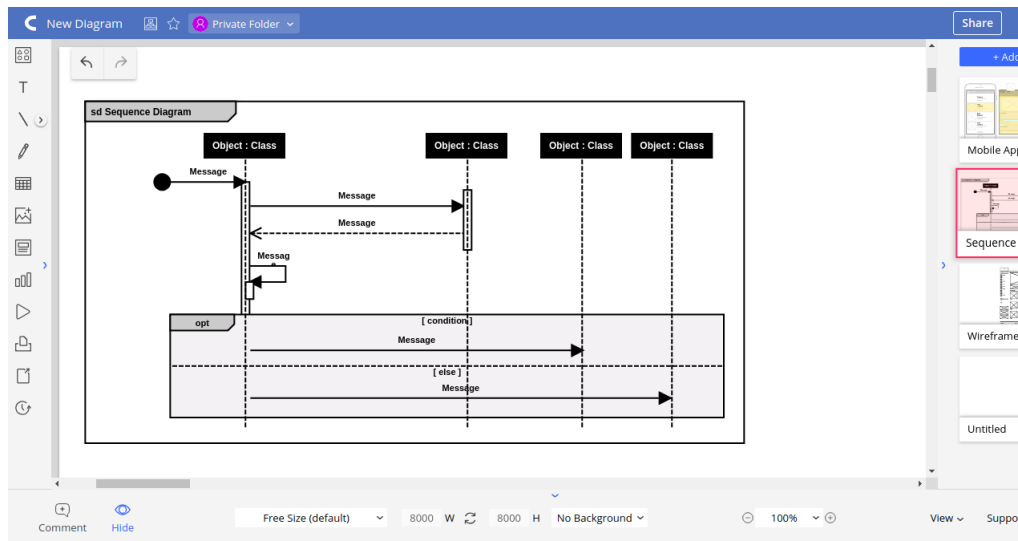
- Plataforma *Web*
- Edição de forma colaborativa.

### 3.4.3 Ferramentas de Análise

Uma vez modelados os requisitos, faz-se necessário conduzir atividades de análise dos mesmos. Sendo assim, foi proposta uma ferramenta voltada para a anotação de alterações no protótipo resultantes da validação do cliente. A Figura 16 ilustra a ferramenta sugerida para construção das anotações resultantes da análise do cliente sobre o protótipo: uma ferramenta *online* de organização de tarefas, chamada *Trello* ([TRELLO, 2019](#)).

#### Especificações da Ferramenta:

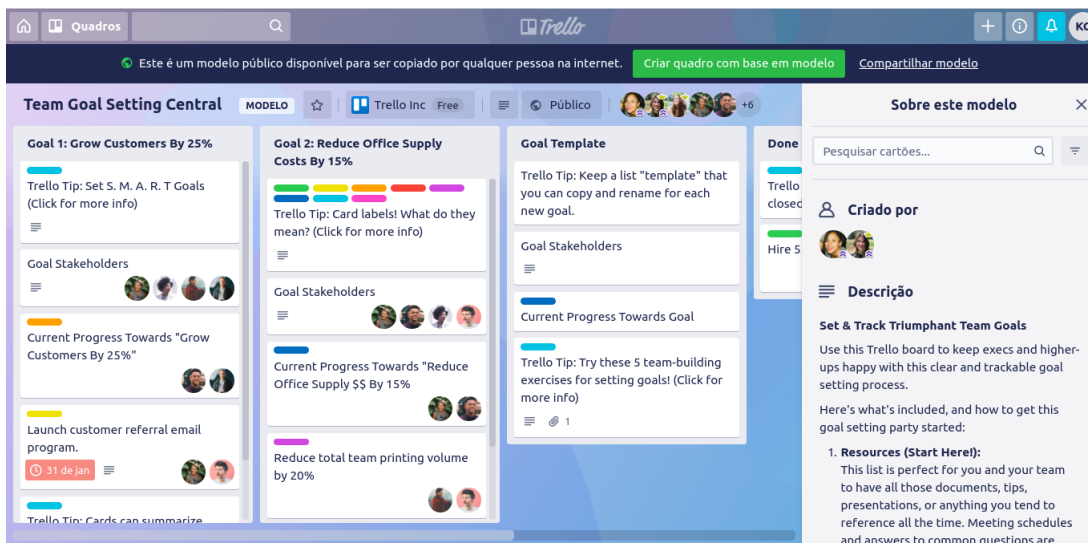
Figura 15 – Ferramenta para elaboração de diagrama de sequência.



Fonte: (CACCO, 2019)

- Plataforma *Web*
- Edição de forma colaborativa, e
- Permite adicionar *labels*, anexos entre outros recursos.

Figura 16 – Ferramenta *trello* para gerenciamento dos resultados de análise do protótipo.



Fonte: (TRELLO, 2019)

### 3.4.4 Ferramentas de Gerência de Requisitos

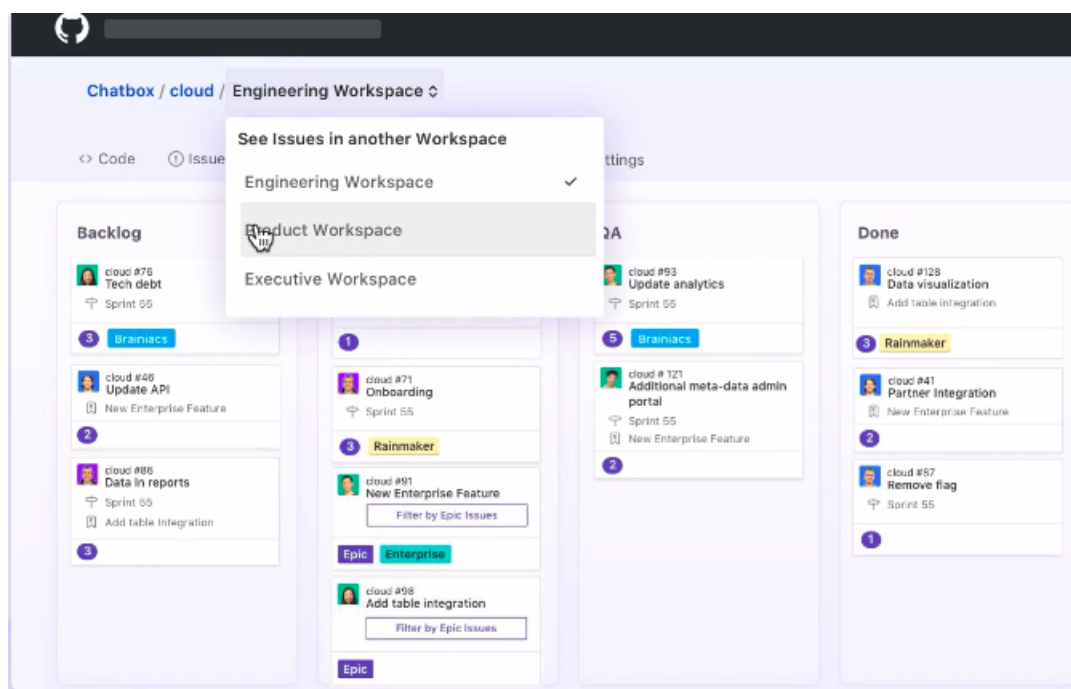
Por fim, em paralelo ao processo cíclico envolvendo as atividades de Elicitação, Modelagem e Análise, tem-se a necessidade de gerenciamento. Nesse caso, foi proposto o uso de uma ferramenta voltada à criação e à categorização do *Backlog* bem como à criação e à pontuação das *issues*.

A Figura 17 ilustra a ferramenta sugerida para elaboração de issues, bem como o registro da pontuação, criação e categorização do *Backlog* : (ZENHUB, 2019), *Plugin* do *Gitub* voltado para o gerenciamento de projetos.

#### Especificações da Ferramenta:

- Plataforma *Web*
- Edição de forma colaborativa, e
- Permite criar e categorizar *Backlog*, adicionar *labels*, assinar membros da equipe e pontuar atividades.

Figura 17 – Ferramenta *zenhub* destinada ao gerenciamento do projeto



Fonte: (ZENHUB, 2019)

## 3.5 Desenho da Abordagem

Para elaboração do desenho referente à abordagem proposta, foi utilizada a ferramenta Bizagi Modeler ([BIZAGI, 2019](#)), em sua versão 3.6. A Ferramenta Bizagi Modeler representa um software livre que tem como propósito a criação e a edição de processos na notação BPMN (*Business Process Modeling Notation*). A seleção desta ferramenta deu-se por questões como: facilidade de uso e capacidade de modelagem dos fluxos de atividades desejados. No Capítulo de [Metodologia](#), constam vários processos que foram desenhados com o apoio desse suporte.

## 3.6 Resumo do Capítulo

Este capítulo buscou apresentar as ferramentas e tecnologias utilizadas para apoiar o desenvolvimento desta abordagem, desde a realização da revisão sistemática, desenho da abordagem e atividades relacionadas à engenharia de software.

Observa-se que as ferramentas utilizadas são em sua maioria gratuitas e, quando possível, *open source*. A análise das ferramentas citadas deu-se com o objetivo de cobrir tanto a parte de documentação e gestão da pesquisa, quanto o desenho da abordagem desenvolvida.

Adicionalmente, destacou-se o uso de ferramentas para apoio nas atividades de elicitação, modelagem, análise e gerenciamento dos requisitos. Lembrando que essas atividades são focos na abordagem desse trabalho.

Por fim, procurou-se documentar o arcabouço tecnológico tipicamente utilizado pelas equipes ágeis, em comunidades de software livre, com destaque para o uso de sistemas operacionais derivados do *Linux*. Portanto, o Estudo de Casos a ser utilizado como base para a aplicação da abordagem foi orientado por esses suportes, mais especificamente: em sistemas operacionais derivados do *Linux*.

## 4 Metodologia

Neste capítulo, são apresentados os procedimentos metodológicos que guiam o presente trabalho em termos mais abrangentes, ou seja: da definição do tema à apresentação aos membros da banca; e em termos mais específicos, com destaque para: [revisão sistemática](#), elaboração da proposta e análise de resultados. Em um primeiro momento, tem-se a [classificação da pesquisa](#), orientando-se por definições da literatura. Em seguida, são cobertas as principais atividades para [condução do TCC](#) como um todo, bem como é acordado um [cronograma](#) para maior detalhamento temporal dessas atividades. Por fim, tem-se o [resumo do capítulo](#).

### 4.1 Classificação da Pesquisa

De acordo com ([GERHARDT; SILVEIRA, 2019](#)), uma pesquisa científica pode ser classificada nas categorias abordagem da pesquisa, natureza da pesquisa, objetivos da pesquisa, e coleta de dados, que são apresentadas nas subseções [4.1.1](#), [4.1.2](#), [4.1.3](#) e [4.1.4](#), respectivamente.

#### 4.1.1 Abordagem da Pesquisa

Baseando-se nos conceitos de ([GERHARDT; SILVEIRA, 2019](#)), a pesquisa foi definida como uma pesquisa *qualitativa*, na qual se preocupa com aspectos da realidade que não podem ser quantificados. [GERHARDT e SILVEIRA \(2019\)](#) relatam ainda que a abordagem *qualitativa* não busca definir resultados de maneira quantificável, mas sim buscando compreender de forma subjetiva o evento estudado, e caracterizando os resultados como satisfatórios ou não.

A pesquisa *qualitativa* tem o ambiente natural como fonte direta de dados e o pesquisador como seu principal instrumento. Segundo ([OLIVEIRA, 2011](#)), a pesquisa *qualitativa* supõe o contato direto e prolongado do pesquisador com o ambiente e a situação que está sendo investigada, via de regra, por meio do trabalho intensivo de campo. O uso da descrição qualitativa procura captar não só a aparência do fenômeno como também suas essências, buscando explicar suas origem, relações e mudanças, e tentando intuir as conseqüências.

A utilização desta abordagem, durante este trabalho teve como objetivo, entender detalhadamente o funcionamento de diversas atividades do processo de engenharia de

software, viabilizando a adaptação das mesmas para um contexto ágil com duração de uma *sprint*.

### 4.1.2 Natureza da Pesquisa

A classificação deste trabalho em relação à natureza da pesquisa é definida como pesquisa *aplicada*, devido ao fato de buscar solucionar um problema recorrente no contexto.

### 4.1.3 Objetivos da Pesquisa

A classificação deste trabalho em relação aos objetivos da pesquisa é definida como *exploratória*. Enquadram-se na categoria dos estudos exploratórios todos aqueles que buscam descobrir idéias e intuições, na tentativa de adquirir maior familiaridade com o fenômeno pesquisado. Nem sempre há a necessidade de formulação de hipóteses nesses estudos. Eles possibilitam aumentar o conhecimento do pesquisador sobre os fatos, permitindo a formulação mais precisa de problemas, criar novas hipóteses e realizar novas pesquisas mais estruturadas. Nesta situação, o planejamento da pesquisa necessita ser flexível o bastante para permitir a análise dos vários aspectos relacionados com o fenômeno.

O autor (GIL, 2008) considera que a pesquisa exploratória tem como objetivo principal desenvolver, esclarecer e modificar conceitos e idéias, tendo em vista a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores. Segundo o autor, estes tipos de pesquisas são os que apresentam menor rigidez no planejamento, pois são planejadas com o objetivo de proporcionar visão geral, de tipo aproximativo, acerca de determinado fato.

### 4.1.4 Procedimentos

Quanto aos procedimentos descritos por (GERHARDT; SILVEIRA, 2019) de coleta da pesquisa, a mesma foi dividida em duas etapas. Durante a primeira etapa, o procedimento seguido foi de Pesquisa Bibliográfica e Revisão Sistemática. Nesse caso, segundo (GERHARDT; SILVEIRA, 2019), esse procedimento é feito a partir do levantamento de referências teóricas publicadas por meio de livros, artigos, páginas da web, entre outros, e a Revisão Sistemática encontra-se documentada no Apendice A

Já na segunda etapa do trabalho, o procedimento executado foi o de Estudo de Caso. De acordo com a classificação da pesquisa nas diferentes categorias apresentadas anteriormente, a Tabela 3 apresenta, de maneira resumida, a classificação da pesquisa..



Tabela 3 – Classificação da pesquisa

Abordagem da Pesquisa	Natureza da Pesquisa	Objetivos da Pesquisa	Procedimentos de Coleta
Qualitativa	Aplicada	Exploratória	Pesquisa Bibliográfica/ Revisão Sistemática / Estudo de Caso

Fonte: Autor

#### 4.1.4.1 Estudo de Caso

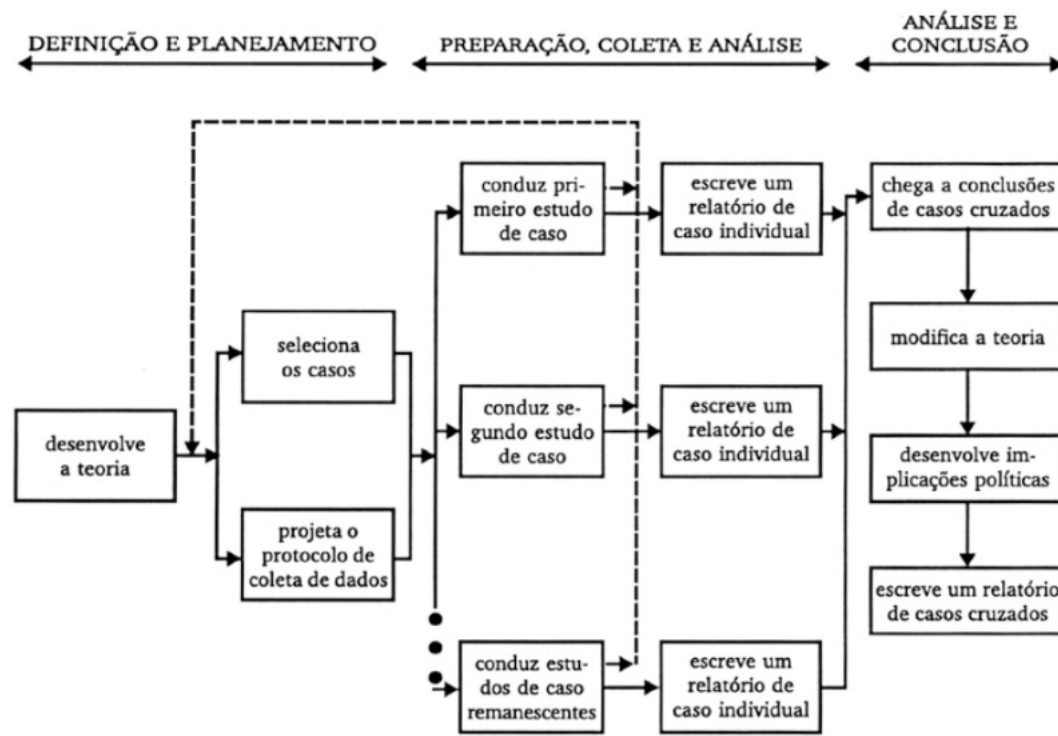
Com o intuito de garantir que as atividades definidas atinjam seus objetivos, será realizado um estudo de caso múltiplo. Segundo (GIL, 2008), o estudo de caso é caracterizado pelo estudo profundo e exaustivo dos fatos objetos de investigação, permitindo um amplo e pormenorizado conhecimento da realidade e dos fenômenos pesquisados.

Martins (2008) define como uma metodologia aplicada para avaliar ou descrever situações dinâmicas em que o elemento humano está presente. Busca-se apreender a totalidade de uma situação e, criativamente, descrever, compreender e interpretar a complexidade de um caso concreto, mediante um mergulho profundo e exaustivo em um objeto delimitado.

O autor (YIN, 2001) afirma que um estudo de caso múltiplo costuma ser mais convincente, e que uma das chaves para se construir um estudo de caso múltiplo bem-sucedido é que este obedeça a uma lógica de replicação, e não a da amostragem, que “exige o cômputo operacional do universo ou do grupo inteiro de respondentes em potencial e, por conseguinte, o procedimento estatístico para se selecionar o subconjunto específico de respondentes que vão participar do levantamento”.

Segundo (YIN, 2001), o desenvolvimento de pode ser esquematizado em forma de fluxograma, de maneira que seja possível categorizar as atividades presentes no fluxo de execução. Adicionalmente, para que seja possível apresentar seus respectivos caminhos de seguimento, conforme representado na Figura 18.

Figura 18 – Fluxograma para um Estudo de Caso Múltiplo



Fonte: (YIN, 2001)

## 4.2 Condução do TCC

Para o desenvolvimento deste trabalho, um processo de condução foi definido. O Trabalho de Conclusão de Curso (TCC), segundo as regras da UnB, é realizado em duas fases: uma chamada de Trabalho de Conclusão de Curso 1 e outra Trabalho de Conclusão de Curso 2. O processo de desenvolvimento do mesmo é representado na Figura 19, conforme desenho esquematizado na ferramenta Bizagi.

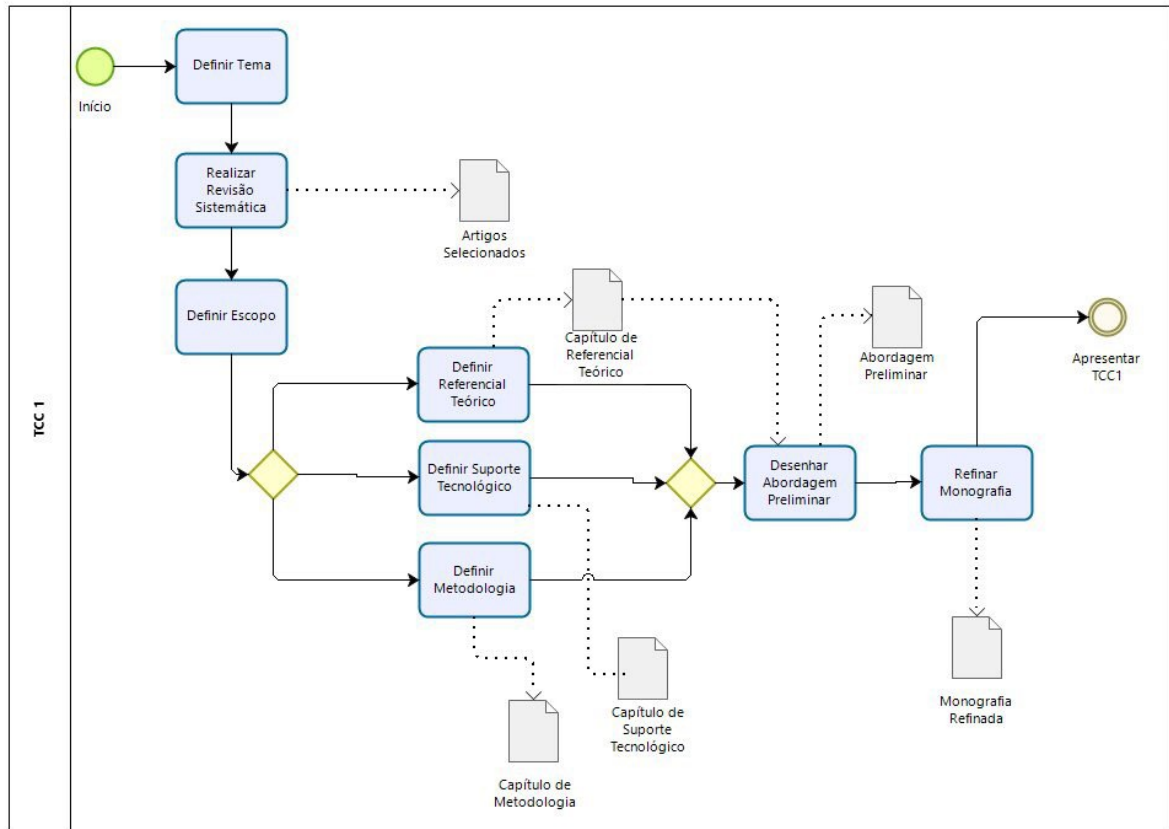
### 4.2.1 TCC1

Durante o TCC 1, o foco desse trabalho foi estabelecer os pilares teóricos para embasamento do projeto como um todo, bem como desenhar uma abordagem preliminar do que foi proposto.

Nessa primeira etapa, o trabalho foi estabelecido considerando nove atividades principais, sendo: selecionar tema, realizar revisão sistemática, definir escopo, escrever referencial teórico, estabelecer suporte tecnológico, definir metodologia, desenhar abordagem preliminar, refinar monografia e apresentar TCC 1, conforme acordado na Figura

19.

Figura 19 – Processo de condução do Trabalho de Conclusão de Curso 1.



Powered by  
bizagi  
Modeler

Fonte: Autor

A seguir, é conferido um maior detalhamento de cada atividade:

- **Definir tema:**

- Descrição: Realizar escolha do tema do trabalho. É necessário que seja realizada com base na percepção de *gap* tecnológico, análise de literatura científica relacionada, afinidade do autor, e possibilidade de pesquisa dentro do tempo da disciplina. Status: concluída;

- **Realizar revisão sistemática:**

- Descrição: Realizar uma revisão sistemática de literatura com o intuito de verificar a existência de pesquisas similares e adquirir conhecimento no tema escolhido. Status: concluída;
- Artefato gerado: Apêndice A, com a Revisão Sistemática realizada.
- **Definir escopo:**
  - Descrição: Delimitação dos limites da pesquisa, com base no tema escolhido. É necessário que seja feita a partir de critérios definidos pelo autor, tais como a relevância da pesquisa para a comunidade, e a não abordagem do tema em trabalhos relacionados. Status: concluída;
- **Definir referencial teórico:**
  - Descrição: Levantar uma fundamentação bibliográfica para o trabalho. É necessário para comprovar que o que está sendo proposto tem real validade. Status: concluída;
  - Artefato gerado: [Capítulo de Referencial Teórico](#).
- **Definir suporte tecnológico:**
  - Descrição: Nesta atividade, são definidas as principais ferramentas e tecnologias utilizadas para a execução deste trabalho. Status: concluída;
  - Artefato gerado: [Capítulo de Suporte Tecnológico](#).
- **Definir metodologia:**
  - Descrição: Definir o tipo de pesquisa a ser feita, podendo incluir detalhes mais específicos sobre o trabalho. Status: concluída;
  - Artefato gerado: [Capítulo de Metodologia](#).
- **Desenhar abordagem preliminar:**
  - Descrição: Desenvolver uma versão preliminar da abordagem. Status: concluída;
  - Artefato utilizado: [Capítulo de Referencial Teórico](#);
  - Artefato gerado: [Capítulo da Abordagem Requirements Sprint](#).
- **Refinar monografia:**
  - Descrição: Refinar o texto do trabalho, que será objeto de avaliação para a disciplina junto à banca. Status: concluída
  - Artefato gerado: Monografia refinada.

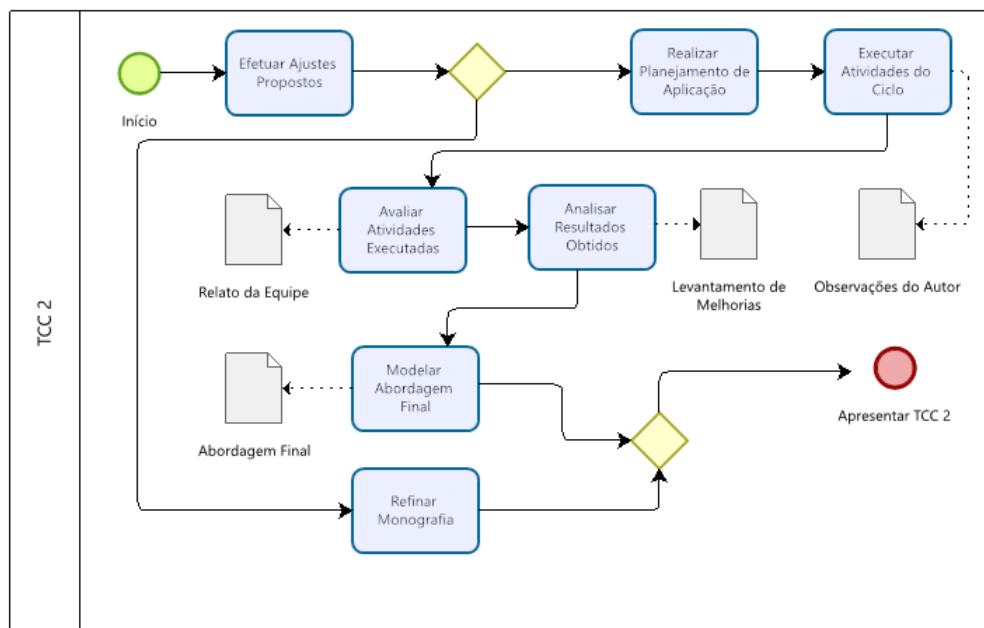
- **Apresentar TCC 1:**

- Descrição: Apresentar para a banca. Marcará o fim da primeira etapa do trabalho.  
*Status:* concluída.

#### 4.2.2 TCC2

Durante o TCC 2, o foco do trabalho foi a aplicação do estudo de caso, bem como a coleta e análise dos resultados obtidos por meio da aplicação do mesmo para aprimoramento da abordagem preliminar denotada anteriormente no TCC1. O TCC2 foi executado por meio de sua aplicação em um estudo de caso, de forma que se tornou possível avaliar a abordagem preliminar, e como consequência, levantar possíveis melhorias, conforme apresentado na Figura 20.

Figura 20 – Processo de condução do Trabalho de Conclusão de Curso 2.



Fonte: Autor

A seguir, é conferido um maior detalhamento de cada atividade:

- **Efetuar ajustes propostos:**

- Descrição: Realizar ajustes propostos pela banca na apresentação da primeira etapa do trabalho (TCC1). *Status*: concluída.
- **Realizar o planejamento da Aplicação:**
  - Descrição: Planejar avaliação. *Status*: concluída;
- **Executar atividades do ciclo:**
  - Descrição: executar às atividades planejadas para o ciclo. *Status*: concluída;
  - Artefato Utilizado: [relatório de planejamento](#);
  - Artefato gerado: Observações do autor com relação às questões levantadas durante a aplicação das atividades propostas na abordagem preliminar;
- **Avaliar atividades executadas:**
  - Descrição: avaliar o resultado do processo e identificar possíveis melhorias. *Status*: concluída;
  - Artefato gerado: [relato da equipe](#);
- **Analisar resultados obtidos:**
  - Descrição: analisar resultados coletados ao longo do ciclo;
  - Artefato utilizado: – Artefato gerado: [relato da equipe](#) e [observações do autor](#);
  - Artefato gerado: [relatório com levantamento de melhorias](#) identificando o estado atual da abordagem com relação à satisfação do processo e possíveis estudos e aprimoramentos futuros;
- **Modelar abordagem final:**
  - Descrição: Modelar e documentar abordagem final, após avaliações. *Status*: concluída;
  - Artefato utilizado: [relatório com melhorias propostas](#);
  - Artefato gerado: [Abordagem final](#);
- **Refinar monografia:**
  - Descrição: Refinar o texto do trabalho, que será objeto de avaliação para a disciplina junto à banca. *Status*: concluída;
  - Artefato gerado: Monografia refinada.
- **Apresentar TCC 2:**
  - Descrição: Apresentar para a banca. Marcará o fim da segunda e última etapa do trabalho. *Status*: em andamento.

## 4.3 Cronogramas

### 4.3.1 Cronograma TCC1

As atividades mencionadas anteriormente, referentes ao TCC1, estão distribuídas de acordo com o cronograma disposto na Figura 21.

Figura 21 – Cronograma de atividades referente ao Trabalho de Conclusão de Curso 1.



Fonte: Autor

### 4.3.2 Cronograma TCC2

As atividades mencionadas anteriormente, referentes ao TCC2, estão distribuídas de acordo com o cronograma apresentado na Figura 22.

Figura 22 – Cronograma de atividades referente ao Trabalho de Conclusão de Curso 2.



Fonte: Autor

## 4.4 Resumo do Capítulo

O capítulo abordou as metodologias de pesquisa utilizadas, bem como o detalhamento da forma como foi conduzido este trabalho. Em decorrência à necessidade principal, que é o desenvolvimento de uma abordagem que une conceitos do método *Design Sprint* e *Ágil* para execução de atividades do processo de engenharia de requisitos com duração de uma *sprint*, buscou-se definir uma metodologia de pesquisa que possibilitasse uma ampla visão do contexto, com a identificação de diversas técnicas e métodos para inserção no desenvolvimento da abordagem.

A pesquisa foi classificada como: *exploratória*, de abordagem *qualitativa*, e natureza *aplicada*. Com relação à metodologia de levantamento bibliográfico, foi utilizado o método de pesquisa bibliográfica, mais especificamente Revisão Sistemática. Já a metodologia de análise de resultados foi Estudo de Casos. Ainda neste capítulo, foram conferidos detalhes referentes às principais atividades que guiaram a realização do trabalho como um todo, como cronogramas de desenvolvimento do trabalho com base nas atividades a serem desenvolvidas.



## 5 Requirements Sprint

Este capítulo é dedicado à apresentação da abordagem *Requirements Sprint*, que por sua vez orienta-se por atividades voltadas para o processo de engenharia de requisitos, conceitos advindos do método *Design Sprint* e da metodologia Ágil. Em um primeiro momento, foi elaborada uma Proposta Preliminar conforme, registrado no [Apêndice B](#), com o intuito de propor uma base metodológica para aplicação de um [Estudo de Caso](#). Como consequência, ocorreu a realização de uma reestruturação da mesma, por meio dos [resultados obtidos](#) durante sua aplicação em um projeto de contexto real.

Com base na reestruturação da abordagem, mediante o levantamento de melhorias, são apresentados seus princípios e objetivos de aplicação. Em seguida, é apresentada uma sugestão de [Composição da Equipe](#) para guiar o processo de forma mais estruturada, bem como o *workflow* proposto para execução das [atividades](#) para um maior detalhamento com relação ao objetivo de entrega de cada referido dia, e sugestões de atividades para viabilizar este objetivo. Por fim, tem-se o [Resumo do Capítulo](#).

### 5.0.1 Questionamentos

Foram levantados alguns questionamentos quanto às atividades propostas para o desenvolvimento da abordagem, os quais são listados a seguir:

- Quais artefatos de saída, mapeados no processo, a equipe pode negligenciar por não achar fundamental para o processo? Como lidar com essa questão?
  - O fato da equipe negligenciar alguns artefatos de saída propostos pela abordagem é um risco que se assume. Porém, a ideia da abordagem é sugerir atividades de forma que seja o mais flexível possível ao ponto da equipe conseguir definir se se encaixa ou não em seu perfil ou de substituir por algum artefato que gere mais valor ao produto em questão.
- Quais artefatos podem ser gerados apenas em situações críticas de desenvolvimento visando dosar o trabalho da equipe? Como identificar que a atividade é mapeada como crítica para o projeto ao ponto de necessitar de uma melhor documentação?
  - Alguns artefatos que demandam maior tempo e dedicação da equipe, e que não agregam tanto valor ao produto, mas sim em relação à rastreabilidade e à memória técnica, serão produzidos apenas em situações críticas, as quais serão definidas

pela própria equipe. Sugere-se o uso de métricas, aplicadas e coletadas ao longo do processo de desenvolvimento, visando estabelecer se uma dada situação é ou não crítica.

### 5.0.2 Desenho da Abordagem *Requirements Sprint*

A Figura 23 apresenta a abordagem final desenhada por meio da ferramenta *Bizagi*, na qual é mapeado inicialmente o fluxo de desenvolvimento da mesma, bem como a sugestão de artefatos gerados em cada um dos dias inseridos na *sprint* dedicada à aplicação da abordagem *Requirements Sprint*. Cabe ressaltar que a abordagem possui um *board* específico, que será apresentado mais à frente, que tem como objetivo guiar o facilitador na condução da abordagem, e possibilitar a centralização dos artefatos produzidos por meio das atividades propostas, permitindo assim que a equipe consiga manter a rastreabilidade do processo junto ao cliente.

É de suma importância esclarecer que a abordagem propõe uma lista de artefatos a serem gerados durante o ciclo com o objetivo de guiar a equipe na tomada de decisão. Entretanto, o processo de definição dos artefatos que agregam valor ao processo e ao produto, conforme a visão da equipe, deve ser realizado de maneira anterior ao início da mesma, para que a equipe possa programar previamente quando cada artefato será produzido dentro da abordagem.

### 5.0.3 Princípios para Aplicação da Abordagem

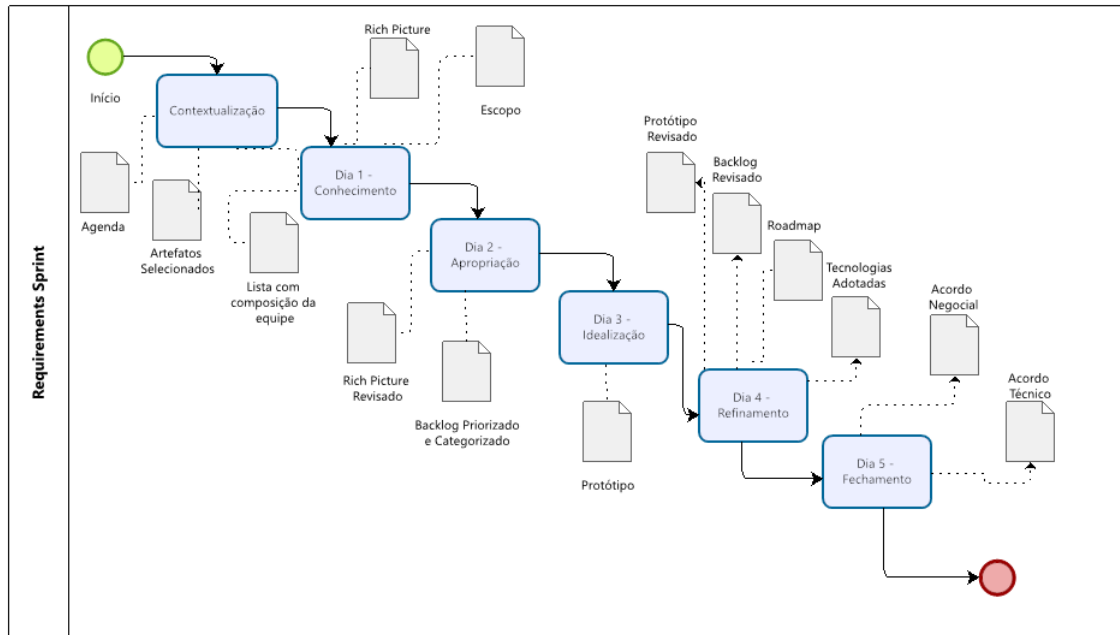
Um dos principais objetivos das atividades presentes na abordagem é que possam ser simples, de modo que os objetivos da execução de cada um dos dias propostos tenham sido concluídos ao final da *sprint*.

Durante a criação das atividades, embora primariamente guiadas às necessidades gerais de composição de produto, foi levado em conta cada princípio aplicável do *Scrum*. A seguir, são apresentados os princípios, junto a identificadores, para que se possa manter uma relação entre estes e as atividades sugeridas.

- **P1 - Empirismo**

Aprendendo com a experiência. Após rodar algumas *sprints* dedicadas à aplicação da *Requirements Sprint* (da maneira correta, executando cada dia proposto pela abordagem), você começará a gerar o seu próprio conhecimento empírico, realimentado a abordagem e aumentando a qualidade e a definição dos melhores artefatos a serem gerados, e que realmente contribuem com o perfil da equipe e do projeto.

Figura 23 – Fluxo da abordagem *Requirements Sprint*



Powered by  
bizagi  
Modeler

Fonte: Autor

- **P2 - Auto-organização**

Quanto maior a alocação de esforço com relação à auto-organização da equipe referente à aplicação das atividades sugeridas pela abordagem, maiores serão os resultados alcançados. Quanto mais auto-organizada for a equipe, maior será a entrega de valor, uma vez que a abordagem conta com uma boa dedicação de entregas e decisões que devem ser concluídas ao final de cada dia.

- **P3 - Colaboração**

Dedica-se às três dimensões básicas referentes ao trabalho colaborativo: consciência, articulação e apropriação, de forma a defender o gerenciamento de projetos como um processo de criação de valor compartilhado, com todas as frentes trabalhando e interagindo em conjunto para atingirem os melhores resultados possíveis por meio da combinação de diferentes visões de produto.

- **P4 - Priorização baseada em Valor**

Neste princípio, destaca-se o máximo de entrega de valor de negócio possível, durante toda execução do projeto. Entregar valor é atender a demanda do seu mercado e/ou resolver o problema do seu cliente, sabendo priorizar da forma correta o que se entende como o sucesso na execução desta atividade.

- **P5 - *Time-boxing***

Os elementos *Time-boxed* referentes à metodologia Scrum incluem as *Sprints*, as Reuniões Diárias, a Reunião de Planejamento da *Sprint*, e a Reunião de Revisão da *Sprint*. Este conceito de *Time-boxed* é aplicado na abordagem para que a equipe alcance a produtividade e consiga ainda melhorar seu comportamento com relação ao princípio anteriormente mencionado, referente à auto-organização. Sendo assim, neste contexto, um princípio contribui com o outro. O princípio de realização de atividades realizadas com prazo e duração limitadas norteia a equipe para que os objetivos levantados sejam atingidos de maneira eficaz e eficiente.

## 5.1 Composição da Equipe

Com relação à composição da equipe para realização da aplicação da abordagem no contexto do projeto por meio da dedicação da equipe pelo período de uma *sprint*, não existe uma exigência para que se obtenha sucesso na inserção da mesma. A abordagem faz uso de sugestões a fim de orientar a equipe quanto a participações relevantes de acordo com o contexto da empresa e sua composição comum.

Neste momento, exige-se uma atenção maior do facilitador para que o mesmo conheça sua equipe de forma a facilitar a tomada de decisão referente à composição da equipe durante o processo de aplicação da abordagem, garantindo assim que estejam presentes os membros que de fato estejam engajados e dispostos a contribuir com o processo como um todo. Fundamentado neste ponto, destaca-se a importância da contextualização prévia da equipe com relação a abordagem que será aplicada, para que os mesmos compreendam o processo que será executado e os ganhos que serão obtidos.

Logo, não existe uma exigência de papéis específicos que devem estar presentes, apenas sugestões para que a empresa possa se apropriar da composição que melhor lhe favorece e que esteja inserida em sua realidade de trabalho atual. Levando em consideração o fato de que a abordagem foi desenvolvida em conjunto com os conceitos advindos do método *Design Sprint*, os seguintes membros são sugeridos para composição da equipe:

- *Designer* (Pessoa responsável por tratar a parte referente ao *design* do que se pretende desenvolver);

- *Product Manager/Product Owner* (Pessoa que conhece a fundo todo produto e seus desafios);
- *Stakeholder* (Parte interessada no negócio);
- Desenvolvedores ou membro com conhecimento técnico (Responsáveis por transformar as ideias advindas do processo inicial de concepção em código), e
- Mediador ou facilitador (Responsável por guiar a equipe para que não se perca o foco, sendo também o responsável por conduzir as sessões coletivas).

## 5.2 Estruturação das Atividades

Além do mapeamento inicial realizado anteriormente, cada atividade possui **Dia**; **Tema**; **Foco**; **Objetivo**; **Definições**; **Atividades**; com seus respectivos **Insumos** e **Artefatos de Saída** sugeridos, assim como o **Princípio do Scrum** trabalhado.

- **Dia**: Possui como objetivo especificar de forma numérica e crescente a que dia se refere determinada atividade;
- **Tema**: Possui como objetivo especificar o tipo do conjunto de atividades que será realizada no determinado dia;
- **Foco**: Possui como objetivo especificar quais atividades de requisitos são tratadas no determinado dia;
- **Objetivo**: Utilizado para especificar o que se pretende alcançar com a execução de atividades propostas no referido dia;
- **Definições**: Possui como objetivo determinar o que ao final do dia se deseja ter definido;
- **Atividades**: Possui como objetivo especificar quais as atividades sugeridas pela abordagem serão realizadas pela equipe para alcance de suas metas;
- **Insumos**: Possui como objetivo especificar o material que servirá de apoio para execução das atividades pela equipe;
- **Artefato de Saída**: Possui como objetivo especificar quais serão os documentos gerados ao final da aplicação de cada atividade proposta, e
- **Princípio do Scrum**: Possui como objetivo especificar qual princípio do scrum está sendo utilizado nas atividades do determinado dia.

### 5.2.1 Apresentação do Workflow da Abordagem

A abordagem *Requirements Sprint* foi criada com a intenção de ser minimalista, eficaz e acima de tudo, ágil. Na Figura 24, é apresentado o *workflow* da abordagem, de forma a possibilitar que a equipe trabalhe de maneira colaborativa na obtenção dos requisitos do produto.

Figura 24 – *Workflow* da abordagem *Requirements Sprint*



Fonte: Autor

Para possibilitar uma condução da abordagem de maneira guiada em que a equipe possa centralizar os registros obtidos durante a execução das atividades, foi elaborado

um *board* dedicado à *Requirements Sprint* para permitir centralizar a comunicação com o cliente e manter a rastreabilidade dos artefatos produzidos.

*Link de acesso ao Board*

### 5.2.2 Descrição da Programação

A programação referente à *sprint* é descrita a seguir, utilizando a estrutura acordada anteriormente.

**Dia:** Anterior ao dia de início da execução da abordagem

**Tema:** CONTEXTUALIZAÇÃO

**Foco:** Apresentação da abordagem aos envolvidos

**Objetivo:** Realizar apresentação da abordagem de maneira que os envolvidos possam compreender o processo e o valor agregado.

**Definições:**

- Agenda de execução da abordagem
- Membros que vão compor a equipe
- Artefatos que serão produzidos

**Atividades:**

Está programada 1 atividade para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado.

**Atividade 1:**

*Apresentação da abordagem para o cliente/stakeholder e os demais membros da equipe*

- Insumos:
  - *Board da abordagem*
  - Documentação da abordagem

- Artefatos de Saída:
    - Documentação das decisões citadas anteriormente podendo, sugestivamente, serem realizadas por meio do preenchimento dos cartões referentes ao dia de contextualização presente no *board da abordagem*
  - Princípio(s) *Scrum*: P2, P3 e P5
- 

**Dia:** 1

**Tema:** CONHECIMENTO

**Foco:** Pré-Rastreabilidade e Elicitação

**Objetivo:** Mapear e entender o problema pesquisando, levantando hipóteses e elaborando ideias.

**Definições:**

- Ideias
- Escopo

**Atividades:**

Estão programadas 2 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados.

**Atividade 1:**

*Briefing* com o cliente/*stakeholder*

- Insumos:
  - Fala do cliente
  - Material disponibilizado pelo cliente
  - Dúvidas que devem ser sanadas durante o *briefing* mediante definição de melhor momento determinada pelo facilitador.



- Artefatos de Saída:
  - Anotações da equipe sobre o *briefing* realizado, podendo, sugestivamente, serem realizadas por meio do preenchimento dos cartões referentes ao dia de conhecimento presente no *board da abordagem*
- Princípio(s) *Scrum*: P2, P3 e P5

### **Atividade 2:**

Debate de ideias com base no *briefing* realizado anteriormente para elaboração do *Rich Picture*.

- Insumos:
  - Anotações da atividade anterior
- Artefatos de Saída:
  - *Rich Picture* elaborado pela equipe, podendo, sugestivamente, ser registrado por meio do preenchimento do cartão referente ao dia de conhecimento presente no *board da abordagem*
- Princípio(s) *Scrum*: P1, P2, P3 e P5

---

**Dia:** 2

**Tema:** APROPRIAÇÃO

**Foco:** Elicitação e Categorização

**Objetivo:** Organizar e categorizar ideias.

**Definições:**

- Apresentação do *Rich Picture*
- Levantamento do *Backlog*
- Priorização e Categorização do *Backlog*

**Atividades:**

Estão programadas 2 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados.

### **Atividade 1:**

Apresentação do *Rich Picture* para o cliente, de forma que um dos membros realize a apresentação do esquema e outro membro faça as anotações com base nas pontuações do cliente para levantar melhorias/correções.

- Insumos:
  - *Rich Picture* elaborado na atividade anterior
- Artefatos de Saída:
  - *Rich Picture* refatorado e acordado com o cliente, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de apropriação presente no *board da abordagem*.
- Princípio(s) *Scrum*: P1,P2, P3 e P5

### **Atividade 2:**

Levantamento, priorização e categorização do *backlog*.

- Insumos:
  - *Rich Picture* acordado anteriormente.
- Artefatos de Saída:
  - Documento com definição de *Backlog* e criação de *labels* referentes à priorização e/ou categorização inseridas ao documento, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de apropriação presente no *board da abordagem*
- Princípio(s) *Scrum*: P1,P2, P3, P4 e P5

**Observações:**

Sugere-se para priorização do *Backlog* o uso da técnica de priorização *Moscow* (MOSCOW, 2019), por ser simples e permitir priorizar em quatro níveis de granularidade, sendo: *Must have*, *Should have*, *Could have* e *Won't have*. Mas, caso a equipe já faça uso de outro recurso para conduzir a priorização, a abordagem é flexível nesse ponto.

---

**Dia:** 3**Tema:** IDEALIZAÇÃO**Foco:** Modelagem**Objetivo:** Prototipação baseada na categorização do *Backlog*.**Definições:**

- *Design*
- Rotas

**Atividades:**

Está programada 1 atividade para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização da atividade seja concluída dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução da atividade, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados e a priorização baseada em valor para se obter sucesso na execução de uma atividade mapeada para este dia.

**Atividade 1:**

Utilização de *wireframes* e/ou *mockups* para construção pareada por *theme* ou *epic*.

- Insumos:
  - *Rich Picture* e *Backlog* levantado anteriormente.
- Artefatos de Saída:

- Protótipo de preferência de alta fidelidade, mas, no mínimo, um protótipo de baixa fidelidade, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de idealização presente no *board da abordagem*.

- Princípio(s) *Scrum*: P1,P2, P3 e P5

**Observações:** Durante o desenvolvimento da prototipação, é natural que a equipe consiga levantar de maneira independente ao cliente pontos de melhoria e/ou correções no backlog, devendo assim realizar o registro quando ocorrer.

---

**Dia:** 4

**Tema:** REFINAMENTO

**Foco:** Modelagem e Análise

**Objetivo:** Validar protótipo com cliente e alinhar o que for necessário para se obter a ideia mais concreta possível do produto com relação à visão e às necessidades do cliente.

**Definições:**

- Protótipo revisado
- *Backlog* revisado
- Criação de um *roadmap* com base no *backlog* e protótipo refinado
- Definição das tecnologias que serão adotadas

**Atividades:**

Estão programadas 3 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades seja concluída dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados e a priorização baseada em valor para se obter sucesso na execução de uma atividade mapeada para este dia.

**Atividade 1:**

Validação do protótipo com o cliente.

- Insumos:
  - Protótipo gerado anteriormente.
- Artefatos de Saída:
  - Anotações pontuais como levantamento de melhorias/correções que devem ser documentadas e acordadas pelo cliente, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de refinamento presente no *board da abordagem*.
  - Protótipo refinado, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de refinamento presente no *board da abordagem*.
- Princípio(s) *Scrum*: P1,P2, P3 e P5

### **Atividade 2:**

Refatoração do *backlog*.

- Insumos:
  - *Backlog* levantado anteriormente.
- Artefatos de Saída:
  - Anotações pontuais como levantamento de melhorias/correções que devem ser documentadas e acordadas pelo cliente, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de refinamento presente no *board da abordagem*.
  - Protótipo refinado, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de refinamento presente no *board da abordagem*.
- Princípio(s) *Scrum*: P1,P2, P3 e P5

### **Atividade 3:**

Elaboração do *roadmap* e definição das tecnologias que serão adotadas durante o processo.

- Insumos:
  - *Backlog* acordado na atividade anterior.

- Artefatos de Saída:
    - Documentos de *roadmap*, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de refinamento presente no *board da abordagem*.
    - Documento de acordo referente as tecnologias que serão adotadas, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de refinamento presente no *board da abordagem*.
  - Princípio(s) *Scrum*: P1, P2, P3, P4 e P5
- 

**Dia:** 5

**Tema:** FECHAMENTO

**Foco:** Rastreabilidade

**Objetivo:** Organizar decisões e firmar acordos com o cliente com base nas decisões e artefatos gerados.

**Definições:**

- Alinhamento técnico e negocial;

**Atividades:**

Estão programadas 3 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados e a priorização baseada em valor para se obter sucesso na execução de uma atividade mapeada para este dia.

**Atividade 1:**

Apresentação do *roadmap* do projeto para cliente.

- Insumos:
  - *roadmap* desenvolvido na atividade anterior.

- Artefatos de Saída:
  - Validação do *roadmap*, podendo haver alterações ou não, de forma a possibilitar acordo com o cliente referente a prazos de entrega.
- Princípio(s) *Scrum*: P1, P2, P3, P4 e P5

### **Atividade 2:**

#### Alinhamento negocial

- Insumos:
  - Elaboração de questionamentos para validação com o cliente do status negocial do produto, podendo, sugestivamente, fazer uso do *template* oferecido nos cartões referentes ao dia de fechamento presente no *board da abordagem*.
- Artefatos de Saída:
  - Registro e validação de todos os artefatos e decisões negociais tomadas durante o processo que foram acordadas junto ao cliente, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de fechamento presente no *board da abordagem*, permitindo assim criar uma rastreabilidade do acordo realizado.
- Princípio(s) *Scrum*: P1, P2, P3, P4 e P5

### **Atividade 3:**

#### Alinhamento Técnico

- Insumos:
  - Elaboração de questionamentos para validação com o cliente do status técnico do produto (caso o cliente possua perfil técnico para participar da etapa de decisões técnicas), podendo, sugestivamente, fazer uso do *template* oferecido nos cartões referentes ao dia de fechamento presente no *board da abordagem*.
- Artefatos de Saída:
  - Registro e validação de todos os artefatos e decisões técnicas tomadas durante o processo que foram acordadas junto ao cliente, podendo, sugestivamente, ser registrado por meio do preenchimento dos cartões referentes ao dia de fechamento

presente no *board* da abordagem, permitindo assim criar uma rastreabilidade do acordo realizado.

- Princípio(s) *Scrum*: P1, P2, P3, P4 e P5

### 5.3 Resumo do Capítulo

Este capítulo expôs a abordagem *Requirements Sprint*. Foi detalhado o processo de execução da abordagem. Esse mapeamento permitiu embasar a visão genérica da mesma, bem como estabelecer um fluxo de atividades existentes, bem como princípios aplicados advindos do *scrum*; a sugestão de composição da equipe para participação do desenvolvimento do processo engenharia de requisitos em cinco dias, e a definição de cada um dos referidos dias e suas devidas atividades sugeridas a serem executadas ao longo da *sprint*. Tem-se ainda os artefatos como meio de sugestão, os quais podem ser gerados ao final de cada atividade exercida com a aplicação da abordagem *Requirements Sprint*.



## 6 Estudo de Caso

No capítulo anterior, *Requirements Sprint*, foi apresentada uma abordagem ágil para possibilitar a aplicação de boas práticas da Engenharia de Requisitos em apenas cinco dias. A abordagem apresenta um olhar ágil, voltado para o âmbito prático, considerando o contexto de *startups* de pequeno e médio portes. Dependendo da empresa, a aplicação da abordagem pode se tornar inviável, sendo necessários ambos, um estudo e uma análise prévia dos processos existentes.

Um estudo de caso foi realizado para que fosse possível simular a inserção da abordagem em um projeto de contexto ágil e real. O estudo de caso teve como objetivo o desenvolvimento de um produto de software com base nos serviços disponibilizados pelo banco central para viabilizar a utilização do PIX, um sistema de pagamentos instantâneo, dentro do contexto de uma *startup* que oferece formas de parcelamento de débitos públicos e privados. Com o início do período de desenvolvimento do produto, foi possível planejar e priorizar as atividades a serem executadas, com base no processo sugerido pela abordagem, e de acordo com cada um dos dias. A ideia era validar a proposta em um contexto real, procurando vivenciar cada iteração, além de manter os rastros quanto aos dados observados.

Esse capítulo está dividido em seções. A seção de [Planejamento](#) apresenta um questionamento, o qual se pretende responder com o Estudo de Caso. Na seção referente a [Seleção do Caso](#), é descrito o cenário de uso, no qual a abordagem foi aplicada. Na seção [Preparação para a Coleta e Análise dos Dados](#), tem-se a preparação para a aplicação da abordagem em si. Na seção de [Coleta e Análise de Dados](#), são apresentadas, respectivamente, a coleta e a análise dos dados obtidos. A seção referente a [Reportar Resultados](#) confere os principais resultados obtidos ao longo desse processo. Esses dados serviram como insumos para a refatoração da abordagem, gerando uma nova versão da mesma, e concluindo o Estudo de Caso. Por fim, tem-se o resumo do capítulo.

### 6.1 Planejamento

Com base no objetivo geral, proposto por esse Trabalho de Conclusão de Curso, apresentado na seção 1.2, foi elaborado um Estudo de Caso, com o intuito de responder o seguinte questionamento: **De que maneira a implementação conjunta de conceitos provenientes das práticas ágeis e do método *Design Sprint* pode contribuir**

para o desenvolvimento de uma abordagem capaz de apoiar a elicitação, a modelagem e a análise de requisitos em empresas e/ou organizações de pequeno ou médio portes?

## 6.2 Seleção do Caso

Um Estudo de Caso é constituído por dois tipos de entidades: casos e unidades de análise. De acordo com a identificação das entidades, o Estudo de Caso pode ser classificado em *holistic*; quando a unidade é o próprio caso em estudo, e em *embedded*; quando mais de uma unidade é atribuída a um caso, segundo (YIN, 2001).

O Estudo de Caso, selecionado para este trabalho, possui um caso e uma unidade de análise. Logo, é classificado na categoria *holistic*.

### Caracterização do Caso

A fintech brasileira escolhida para participar do *Lift Learning*, programa da Fenasbac em parceria com o Banco Central, que busca desenvolver soluções inovadoras que beneficiem o Sistema Financeiro Nacional e a população, frente ao atual cenário de crise na saúde, com repercussão direta na economia em decorrência da Covid-19, no qual foi aplicado o Estudo de Caso, é a Vamos Parcelar. Essa *fintech* possui como propósito possibilitar a quitação de débitos públicos e privados, junto aos órgãos competentes, por meio do parcelamento via cartão de crédito.

**Definição da Unidade Caso** Esse Estudo de Caso foi projetado para que fosse possível a aplicação da abordagem durante o processo de desenvolvimento de implantação de um sistema de pagamentos instantâneos PIX. No início desse trabalho, foram identificados, dentro do ciclo de atividades propostas, quais artefatos retornariam valor para o processo e o produto para que pudesse ser filtrado apenas o necessário.

## 6.3 Preparar para a Coleta e a Análise dos Dados

As técnica de coleta utilizada foi a de observação. A aplicação da abordagem foi realizada considerando o processo de desenvolvimento de software existente, realizado na *fintech* em estudo, e teve como foco o comportamento equipe diante das atividades propostas, com o objetivo de atender as práticas inerentes à fase de requisitos, dentro do escopo do projeto.

A *fintech* Vamos Parcelar possui uma plataforma responsável por possibilitar a quitação de débitos públicos e privados, sejam eles: boletos diversos, água, luz, multas,

IPVA e entre outros, por meio do parcelamento cartão de crédito. Esse contexto motivou sua participação no projeto Lift Learning, tendo como objetivo expandir os meios de pagamento da plataforma por meio da disponibilização da quitação de débitos via PIX. Assim, torna-se possível acompanhar o mercado financeiro e suas inovações de pagamento, uma vez que essa *fintech* tem por meta se tornar uma carteira digital de referência para seus atuais e futuros clientes.

### 6.3.1 Composição da Equipe

A equipe envolvida no processo foi composta por 8 (oito) desenvolvedores Juniores, 2 (dois) desenvolvedores plenos e a parte interessada da empresa inserida no contexto, no qual estiveram presentes um Scrum Master e um arquiteto de software da empresa Vamos Parcelar.

A abordagem preliminar propõe a participação de 1 *design*, 1 *product owner*, 1 *stakeholder*, 1 facilitador e desenvolvedores envolvidos com o produto. Entretanto, foi necessária uma readequação à realidade, em particular no que tange a disposição dos envolvidos. Como a abordagem confere essa flexibilidade, optou-se por uma adaptação, sendo essa composta por:

- 1 *Scrum Master*: Realizando os papéis do *Stakeholder* e do *Design* por possuir uma boa visão negocial do produto e ser cliente da plataforma, conseguindo, portanto, levantar bem as reais necessidades da empresa;
- 1 Arquiteto de Software: Realizando o papel do *Product Owner*, devido ao fato de ter uma abrangente visão do produto de maneira técnica e negocial, de forma a facilitar o levantamento de prioridades dentro do escopo a ser levantado;
- 2 Desenvolvedores Plenos: Realizando o papel de Facilitador do Processo, e
- 8 Desenvolvedores Juniores: Realizando a abstração das ideias advindas do processo de concepção inicial em código.

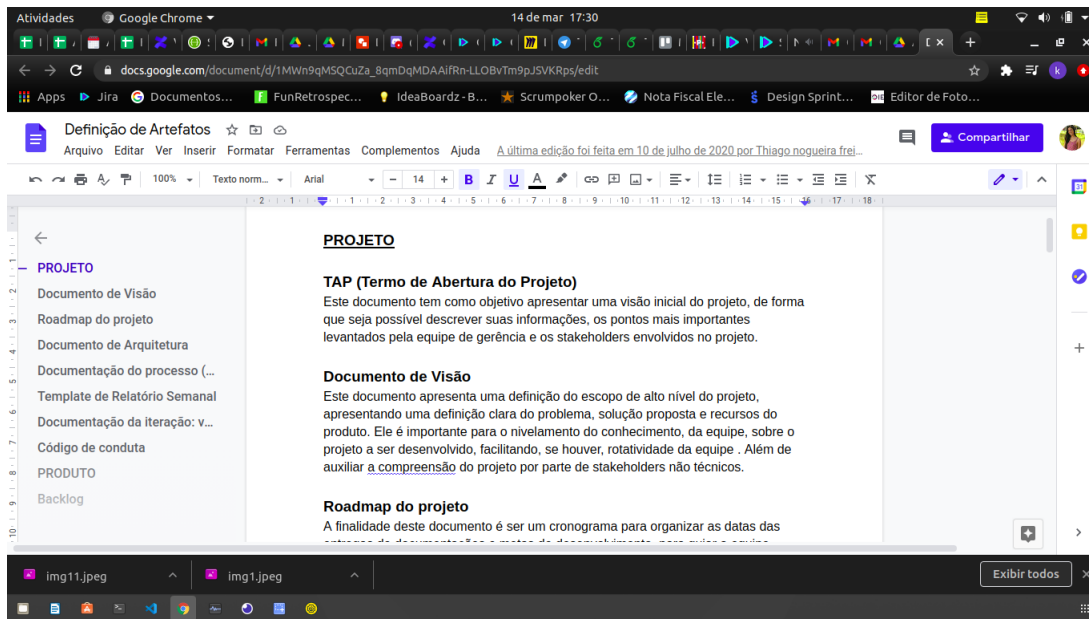
### 6.3.2 Contextualização Prévia da Abordagem para Priorização de Artefatos

Inicialmente, para possibilitar a aplicação da abordagem no contexto do projeto, foi realizada uma apresentação da proposta para a equipe referente aos artefatos a serem produzidos, com o objetivo de que a equipe pudesse compreender o valor agregado, e permitindo, assim, a participação ativa e colaborativa dos envolvidos. A ideia era coletar pontos de melhoria no processo da abordagem preliminar.

Com isso, foi sugerida, após uma análise criteriosa do processo por todos os membros da equipe, a realização de um levantamento, de acordo com as sugestões da abordagem, dos artefatos que de fato agregariam valor ao projeto e, desta maneira, refinar o estudo de caso com base nos levantamentos realizados. Desta forma, foi realizado, entre os desenvolvedores plenos e juniores, um levantamento dos artefatos e suas respectivas justificativas. O intuito é coletar justificativas, orientando-se por uma visão mais técnica, e permitindo perceber o que agregaria mais valor documental para o produto, uma vez que o mesmo se trata de um software livre e disponível para contribuição.

A Figura 25 apresenta um artefato documental que procura acordar os principais artefatos a serem utilizados pela equipe na realização da abordagem bem como as justificativas apontadas pelos desenvolvedores plenos e juniores. Por exemplo, quanto ao Termo de Abertura do Projeto, a justificativa compreende "Este documento tem como objetivo apresentar uma visão inicial do projeto, de forma que seja possível descrever suas informações, os pontos mais importantes levantados pela equipe de gerência e os stakeholders envolvidos no projeto."

Figura 25 – Lista de artefatos e justificativa de utilização



Fonte: Autor

## 6.4 Coleta e Análise de Dados

A elaboração da abordagem foi construída considerando sua execução dentro do prazo de cinco dias, levando em consideração o método *Design Sprint* que possui dura-

ção de uma semana. Após a elaboração desse modelo, foi possível refiná-lo por meio de observações realizadas mediante as interações dos atores com a abordagem, durante seu período de aplicação.

Após serem acordados os artefatos a serem produzidos durante o Estudo de Caso, foi dado início à execução da abordagem com o desenvolvimento das atividades propostas previamente, conforme especificado a seguir.

### 6.4.1 Execução da Programação

A abordagem sugere que a equipe realize um primeiro ciclo com base nas atividades propostas e avalie a necessidade de inserção de um ou mais ciclos de execução da abordagem para refinamento ou refatoração dos requisitos levantados no ciclo anterior.

#### 6.4.1.1 Primeiro Ciclo

Anteriormente ao início do ciclo, foi acordada, junto à empresa, a aplicação da abordagem, no intuito de realizar o levantamento dos requisitos do projeto. Foi encaminhado um esquema preliminar da abordagem para que a empresa tomasse conhecimento da metodologia que seria utilizada. Adicionalmente, esse passo possibilitou a criação de uma agenda, junto aos envolvidos nas atividades, e permitiu, na sequência, organizar o tempo orientando-se pelas disponibilidades informadas.

**Dia 1:** Conhecimento

**Data da realização:** 04/08/2020

**Definições:**

- Ideias
- *Rich Picture*

**Atividades:**

**Atividade 1:**

- Envolvidos: (1 *Scrum Master*, 2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
  - Descrição do Processo: Foi realizado um *briefing*, no qual o time já estava alinhado com as expectativas do projeto e possuía como objetivo compreender melhor a empresa e suas necessidades com relação ao produto a ser desenvolvido. Com isso, a empresa Vamos Parcelar, realizou uma apresentação do produto atual, bem como suas expectativas referentes ao valor incremental que a inserção do PIX traria para a plataforma existente. Além

disso, foram abordados assuntos referentes a processos e metodologias de desenvolvimento executados internamente na empresa.

### **Atividade 2:**

- Envolvidos: (2 Desenvolvedores Plenos)

- Descrição do Processo: Para fechar as atividades propostas para o primeiro dia de interação, foi desenvolvido um *Rich Picture* de forma colaborativa entre o time, com o intuito de levantar uma proposta de escopo para ser posteriormente apresentado para a empresa.

O *Rich Picture* apresenta o escopo preliminar levantado após realização do *briefing*, o qual foi utilizado para possibilitar acordar um escopo junto à empresa de maneira mais negocial. O esquema a seguir especifica o fluxo que o usuário percorre ao relizar um pagamento dentro da plataforma e o retorno que o mesmo obtém ao finalizar a atividade.

---

**Dia 2:** Apropriação

**Data da realização:** 05/08/2020

**Definições:**

- Escopo

- *Backlog*

**Atividades:**

### **Atividade 1:**

- Envolvidos: (1 *Scrum Master*, 2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)

- Descrição do Processo: Nesta atividade, o time levou o *Rich Picture* elaborado anteriormente, conforme proposto pela abordagem. Esse artefato, então, foi apresentado e melhorado de maneira incremental, durante a apresentação realizada, e de acordo com as expectativas e possibilidades da empresa. Ao final da reunião, o escopo havia sido alinhado e acordado entre todos os membros do time e a empresa.

### **Atividade 2:**

- Envolvidos: (2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)

- Descrição do Processo: Foi realizado o levantamento com base no *Rich Picture* acordado na atividade anterior, de um *Backlog* de forma macro, composto por *features* e priorizações com base no processo evolutivo de desenvolvimento do produto e suas dependências. Para isso, foi utilizado como apoio ao levantamento do *Backlog* a ferramenta *Clickup*.

O *Backlog preliminar* especifica a forma como foi construído após a validação do *Rich Picture*, no qual foram utilizadas *tags* visando identificar os *epics*, associando-os às *features*.

---

### **Dia 3:** Idealização

**Data da realização:** 06/08/2020

#### **Definições:**

- Elaboração do Protótipo
- Políticas de Contribuição

#### **Atividades:**

##### **Atividade 1:**

- Envolvidos: (2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
- Descrição do Processo: Com base no *Rich Picture* e no *Backlog* do produto, foi desenvolvida uma prototipação utilizando a ferramenta *Figma*, bem como definidas as políticas de contribuição para o projeto a serem adotadas pelos desenvolvedores para possibilitar criar um padrão de contribuição no projeto.

O *protótipo preliminar* apresenta o esboço da aplicação a ser desenvolvida por meio da utilização da ferramenta *Figma*.

---

### **Dia 4:** Refinamento

**Data da realização:** 07/08/2020

#### **Definições:**

- Refinamento do Protótipo
- Refinamento do *Backlog*

- Definição de Tecnologias
- Termo de Abertura do Projeto
- Documento de Visão
- *Roadmap* do produto

**Atividades:****Atividade 1:**

- Envolvidos: (1 *Scrum master*, 1 Arquiteto, 2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
  
- Descrição do Processo: Foi realizada uma apresentação do protótipo e *Backlog* junto à equipe e à empresa, possibilitando um refinamento colaborativo. Nesta atividade, que envolveu a equipe e a empresa, foi definido, com o apoio do arquiteto, reaproveitar alguns serviços da empresa, como por exemplo: sistema de *login*; listagem de débitos; entre outros. A intenção era reduzir a carga de trabalho e acelerar o processo de entrega da funcionalidade de pagamento via PIX, foco do projeto. Além disso, foram levantadas as tecnologias a serem adotadas para desenvolvimento do software. Essas tecnologias foram sugeridas pela equipe, e validadas pela empresa, com base na facilidade para se reaproveitá-las no futuro.

**Atividade 2:**

- Envolvidos: (2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
  
- Descrição do Processo: Os desenvolvedores plenos trabalharam na elaboração do *roadmap* preliminar do projeto, e os desenvolvedores juniores dividiram-se para produzir o do documento de termo de abertura do projeto, o documento de visão, e o documento de gerenciamento e configuração de software para documentar o *gitflow*, padrões de contribuição e tecnologias adotadas.

O *Roadmap preliminar* apresenta o levantamento realizado com base no prazo para finalização do projeto e *Backlog*.

---

**Dia 5:** Fechamento

**Data da realização:** 08/08/2020

**Definições:**



- Alinhamento Técnico
- Validação do TAP e Documento de Visão
- Documento de Arquitetura

**Atividades:****Atividade 1:**

- Envolvidos: (1 *Scrum master*, 1 Arquiteto, 2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
- Descrição do Processo: Foi realizada uma reunião para fechamento dos requisitos e alinhamento técnico referente à arquitetura adotada, junto ao Arquiteto da Vamos Parcelar, para guiar a elaboração do documento de arquitetura. Esse documento, juntamente com o *roadmap* preliminar, orientam aspectos arquiteturais e de projeto.

**Atividade 2:**

- Envolvidos: (2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
- Descrição do Processo: Foi realizado o envio das documentações desenvolvidas anteriormente, como termo de abertura de projeto e documento de visão, no intuito da empresa realizar uma revisão e, posteriormente, caso necessário, refatorações pontuais.

O [Esquema Arquitetural](#) refere-se ao esboço da arquitetura adotada para o software a ser desenvolvido.

### 6.4.2 Segundo Ciclo

Ao final do primeiro ciclo, foi possível fechar o escopo, levantar e priorizar os requisitos de maneira macro, prototipar, definir as tecnologias utilizadas e produzir alguns artefatos negociais devido ao projeto estar vinculado ao banco central, e ter um cunho voltado ao sistema financeiro, sendo ainda aberto a contribuições. Durante o processo de desenvolvimento, a empresa não conseguiu disponibilizar os serviços que seriam reaproveitados no prazo para que não comprometesse o desenvolvimento do MVP. Com isso, foi necessário rever o produto acordado, para que fosse possível retirar os impedimentos existentes quanto aos serviços que seriam consumidos.

Em decorrência disso, foi realizado um novo levantamento por meio da aplicação de mais um ciclo da abordagem, de maneira a possibilitar o levantamento do escopo com base em um produto desenvolvido de forma independente da Vamos Parcelar. Nesse

contexto, foi necessário o desenvolvimento do *mock* dos serviços que seriam reaproveitados anteriormente. Logo, foi definido que não seria necessário realizar todas as etapas da abordagem novamente, apenas as que fossem realmente agregar valor ao complemento do escopo com os requisitos faltantes.

### **Dia 1:** Conhecimento

**Data da realização:** 31/08/2020

#### **Definições:**

- *Rich Picture*

#### **Atividades:**

##### **Atividade 1:**

- Envolvidos: (2 Desenvolvedores Plenos e 1 Gerente de Projetos)

- Descrição do Processo: Foi realizado o levantamento interno de uma nova proposta de escopo de forma a abranger os serviços que seriam consumidos anteriormente por meio do desenvolvimento de serviços de *mocks* que seriam responsáveis por gerar a chave PIX e retornar um *payload* com os dados para geração do Qrcode, bem como desenvolvimento de funcionalidades que seriam reaproveitadas do sistema existente na empresa.

O *Rich Picture final* foi desenvolvido após levantamento interno para alteração de escopo, utilizado para possibilitar novo acordo junto à empresa, e de maneira negocial. O mesmo apresenta um escopo mais amplo, com a adição dos serviços que anteriormente seriam reaproveitados e que foram adicionados por meio de *mocks*.

---

### **Dia 2:** Apropriação

**Data da realização:** 01/09/2020

#### **Definições:**

- Escopo

- *Backlog*

#### **Atividades:**

##### **Atividade 1:**

- Envolvidos: (1 *Scrum Master*, 2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)

- Descrição do Processo: Nesta atividade, o time levou o *Rich Picture* elaborado anteriormente para realização de um reatendimento de escopo, o qual foi apresentado e melhorado de maneira incremental durante sua apresentação. O novo escopo foi aprovado, e foi tomada a decisão de alteração do desenvolvimento de uma aplicação *mobile* para uma plataforma *web*.

### **Atividade 2:**

- Envolvidos: (2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)

- Descrição do Processo: Foi realizada uma refatoração e recategorização do *Backlog* existente de forma a adicionar ao mesmo a implementação dos *mocks* que substituem os serviços que seriam disponibilizados para consumo pela empresa de forma a possibilitar a geração do Qrcode na plataforma para pagamento, bem como as funcionalidades que seriam reaproveitadas do sistema existente na empresa, que são fundamentais para a triagem do pagamento do cliente via PIX na plataforma, como por exemplo: *login*, cadastro de débitos, entre outros.

O *Backlog* foi reestruturado após validação do *Rich Picture*, inserindo novos requisitos e alterando a ferramenta utilizada para o *Zenhub* com o objetivo de centralizar a documentação e o código em uma mesma plataforma, o *GitHub*.

---

**Dia 3:** Idealização

**Data da realização:** 02/09/2020

**Definições:**

- Elaboração do Protótipo

**Atividades:**

### **Atividade 1:**

- Envolvidos: (2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)

- Descrição do Processo: Com base no *Rich Picture* e no *Backlog* do produto refatorado anteriormente, foi desenvolvido o protótipo de uma plataforma *web* abrangendo as principais funcionalidades que permitem ao usuário ser triado para possibilitar o pagamento de débitos via PIX dentro da plataforma.

O **protótipo final** apresenta a prototipação da plataforma web a ser desenvolvida por meio da utilização da ferramenta *Figma*.

---

#### **Dia 4:** Refinamento

**Data da realização:** 03/09/2020

#### **Definições:**

- Validação do *Backlog* e Protótipo
- Refatoração dos artefatos segundo alterações no escopo.

#### **Atividades:**

##### **Atividade 1:**

- Envolvidos: (1 *Scrum Master*, 1 Arquiteto, 2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
- Descrição do Processo: Foi realizada uma reunião para apresentação do novo *Backlog* recategorizado e do protótipo alterado para que fosse possível validar as alterações junto à empresa e coletar sugestões de melhorias a serem implementadas.

##### **Atividade 2:**

- Envolvidos: (2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)
  - Descrição do Processo: Diante da alteração do escopo, foi levantada a necessidade de refatoração dos artefatos gerados anteriormente. Sendo assim, a equipe dividiu-se para que fosse possível verificar cada um dos documentos e encaminhar para que a empresa pudesse validar as alterações realizadas.
- 

#### **Dia 5:** Fechamento

**Data da realização:** 04/09/2020

#### **Definições:**

- Reunião para alinhamento técnico e comercial

**Atividades:****Atividade 1:**

- Envolvidos: (1 *Scrum Master*, 1 Arquiteto, 2 Desenvolvedores Plenos e 8 Desenvolvedores Juniores)

- Descrição do Processo: Mediante alterações nos artefatos gerados anteriormente, foi necessário realizar uma reunião para alinhamento técnico, referente a arquitetura e tecnologias adotadas, e negocial, com relação aos artefatos e *roadmap* gerados, com o intuito de realizar um acordo final com a empresa para dar início ao desenvolvimento do produto.

---

## 6.5 Reportar Resultados

Conforme descrito anteriormente, foi possível obter ganhos significativos por meio da aplicação da abordagem no levantamento dos requisitos do projeto, abrangendo as atividades do processo de Engenharia de Requisitos de forma ágil e se adequando às reais necessidades da equipe e do projeto. Conforme mencionado nos capítulos anteriores, a pesquisa foi classificada como qualitativa, e em decorrência disso, não busca definir resultados de maneira quantificável e sim de maneira a compreender os fatos e caracterizar os resultados como satisfatórios ou não.

Durante o início de inserção da abordagem no processo de desenvolvimento do projeto, o time mostrou-se motivado devido ao fato de não terem conhecimento de um processo bem definido para a etapa de requisitos e, principalmente, para o levantamento de escopo de maneira guiada junto ao cliente, o que gerou de certa forma uma segurança maior para o time na execução do processo.

Para que fosse possível realizar a coleta de dados resultantes da aplicação da abordagem, foi utilizada a técnica de observação, na qual consiste em anotações realizadas durante o ciclo, mediante o comportamento da equipe durante o processo, bem como a utilização de um questionário para obtenção de dados referentes à satisfação da equipe e da empresa, para que fosse possível levantar sugestões de melhoria para evolução da abordagem e entender até que ponto sua aplicação é relevante para o processo de desenvolvimento do produto.

### 6.5.1 Clareza da Abordagem

Durante o processo de execução, foi priorizado o entendimento e a compreensão de todos os envolvidos, com relação à geração de valor de cada uma das atividades propostas pela abordagem, de forma a permitir que os mesmos pudessem realizar uma avaliação crítica acerca dos artefatos que de fato agregariam valor para o produto, seja ele de compreensão de escopo ou documental.

Com isso, foi proposto pelos próprios membros da equipe, antes de iniciar o ciclo de execução da abordagem, a realização de um levantamento de todos os artefatos propostos em conjunto quanto às justificativas pelas quais determinado artefato agregaria valor para o projeto. Desta forma, foram listados e justificados os artefatos, e posteriormente foi realizada uma reunião para compartilhamento de opiniões e priorização dos artefatos que de fato iriam ser utilizados ou inseridos na abordagem, como foram os casos do documento de visão, do documento de arquitetura, e do documento de gerência de configuração e evolução de software.

Os artefatos acordados para desenvolvimento foram:

- *Rich Pictures*
- *Backlog*
- Documento de Metodologia Adotada
- Protótipo
- Plano de Gerência e Configuração de Software
- Termo de Abertura do Projeto
- Documento de Visão
- *Roadmap* do Produto
- Documento de Arquitetura

### 6.5.2 Abrangência da Abordagem

Por meio da aplicação da abordagem no contexto do projeto, foi possível obter ganhos tanto no levantamento de escopo do produto a ser desenvolvido, quanto na abrangência das atividades de Engenharia de Requisitos e desenvolvimento documental da parte comercial do processo.

Introduzindo a abordagem, foi possível, além de alcançar resultados em termos de geração de artefatos, facilitar a comunicação com o cliente de forma a criar um processo de interação dinâmico e simples, em que a equipe participa de maneira colaborativa e confortável por compreender o processo e o valor que a abordagem agrega. A Documentação técnica e negocial desenvolvida no decorrer da aplicação da abordagem encontra-se registrada na plataforma *Github* por meio do *link*: <<https://github.com/Vamos-Parcelar-Lift-Learning/Documentos>>

### 6.5.3 Flexibilidade da abordagem

Conforme colocado anteriormente, um dos principais pontos da abordagem é a questão da flexibilidade oferecida ao time, que permite que o mesmo defina quais os artefatos relevantes, assim como os integrantes envolvidos no processo de aplicação da abordagem. A abordagem faz sugestão de alguns papéis interessantes a serem envolvidos no processo. Entretanto, o mesmo é adaptável e flexível o suficiente para que seja possível aplicá-lo de acordo com a atual situação da empresa.

No caso da Vamos Parcelar, o quadro de integrantes era limitado. Entretanto, tal fator não foi um impedimento para aplicação e sucesso da abordagem. É possível realizar adaptações, nas quais um ou mais membros assumem mais de um papel ou o que julgarem necessário.

### 6.5.4 Agilidade da Abordagem

A agilidade no processo de aplicação da abordagem é um dos fatores primordiais para que seja possível obter sucesso em sua implementação. Com relação a algumas observações realizadas durante o ciclo, foi possível identificar alguns pontos, entre eles, a construção do protótipo que foi realizada de maneira colaborativa utilizando a ferramenta *Figma*, com a qual os integrantes iam discutindo particularidades do sistema que serviram de insumos para o levantamento de requisitos no *Backlog* que não haviam sido mapeados anteriormente.

Outro ponto levantado é a questão de entregar o mínimo e melhorar de maneira gradual. Sendo assim, todo e qualquer artefato foi produzido a partir dos insumos disponíveis e entregues dentro do ciclo, mesmo que ainda não estivessem em sua melhor versão. Como consequência, essas versões preliminares iam sendo aprimoradas durante o processo de desenvolvimento, sempre que o time identificava essa necessidade.

### 6.5.5 Posicionamento dos Envolvidos sobre a Abordagem

Ao final da aplicação da abordagem no processo, foi levantado um questionário utilizando a ferramenta do *Google Forms*, viabilizando levantamento mais criterioso sobre como os aplicadores da abordagem enxergaram sua implementação. Ao todo, participaram 12 pessoas no processo de aplicação da abordagem, sendo: 1 arquiteto, 1 *Scrum Master*, 2 desenvolvedores plenos e 8 desenvolvedores juniores. Diante deste cenário, foi possível contar com a resposta de 9, das 12 pessoas envolvidas no projeto. Essa amostra de 9 pessoas participou colaborativamente do questionário levantado.

O principal objetivo da implementação do questionário para a equipe é possibilitar que a mesma possa levantar sua experiência particular com a abordagem de maneira confidencial, expondo assim sua visão e opinião com relação ao valor agregado ao projeto.

A seguir, tem-se o registro do questionário, bem como o resultado obtido por meio da aplicação do mesmo ao final do ciclo de interação.

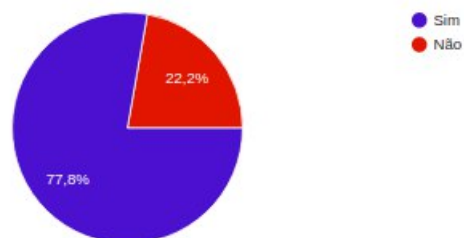
A Figura 26 o questionamento referente ao conhecimento preliminar dos participantes quanto a abordagens e processos atuantes na etapa de requisitos, ou seja, desses participantes conhecerem a implementação da abordagem *Requirements Sprint*.

Foi possível identificar que 77,8% dos participantes declararam já conhecerem abordagens e processos para atuar na etapa de requisitos. Já 22,2% dos participantes afirmaram que não tinham esse conhecimento prévio. Sendo assim, na amostra consultada, há predomínio de participantes que tinham conhecimento prévio sobre o uso de abordagens e processos atuantes na etapa de Engenharia de Requisitos.

Figura 26 – Questão 1 do Questionário de Avaliação

Você conhece ou já utilizou alguma abordagem ou processo para levantamento da camada de requisitos?

9 respostas



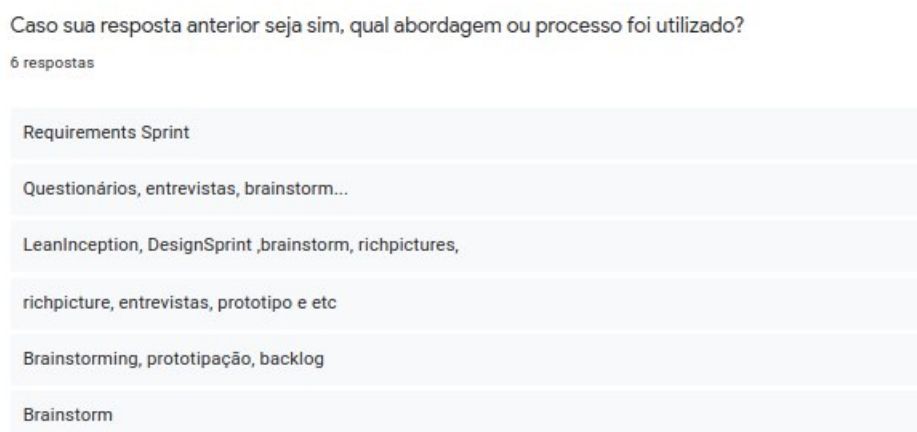
Fonte: Autor

Na sequência, foi solicitado, conforme apresentado na Figura 27, a descrição da abordagem, na qual a equipe tinha conhecimento antes de aplicar a *Requirements Sprint*.



Desta forma, o processo mais conhecido e/ou utilizado durante o levantamento da etapa de requisitos foi o *brainstorming*, e o *Rich Picture* foi o artefato mais útil. Cabe ressaltar que *Rich Picture*, por exemplo, é um dos artefatos propostos nos ciclos de atividades da abordagem *Requirements Sprint*.

Figura 27 – Questão 2 do Questionário de Avaliação



Fonte: Autor

Na Figura 28, foi questionada sobre a clareza na compreensão da execução do processo da abordagem, de forma a ser possível identificar que o mesmo está descrito de maneira objetiva segundo a avaliação da equipe. Como resultado, 100% dos participantes na pesquisa comentam que sim, a abordagem é clara.

Figura 28 – Questão 3 do Questionário de Avaliação



Fonte: Autor

Na sequência, a Figura 29 apresenta o quão confortável a equipe se sentiu ao aplicar a abordagem durante o processo. Como resultado, 100% dos participantes na pesquisa responderam que sim, ou seja, se sentiram confortáveis.

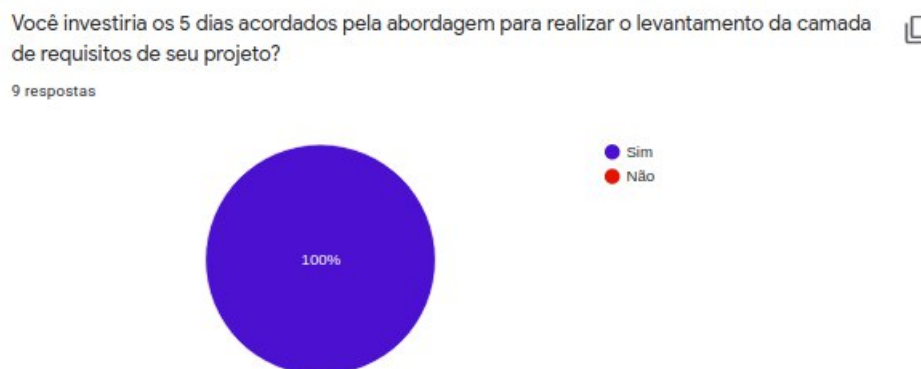
Figura 29 – Questão 4 do Questionário de Avaliação



Fonte: Autor

A Figura 30 ilustra sobre o questionamento se a equipe realmente investiria os 5 (cinco) dias por considerar que os mesmos geram um retorno de valor para o projeto, uma vez que o processo sugerido é capaz de levantar um escopo e cuidar de boa parte das atividades inerentes à Engenharia de Requisitos. Novamente, 100% dos respondentes colocaram que sim. Esse resultado é, particularmente, interessante, pois uma das maiores dificuldades de uma abordagem é a mesma ser reconhecida pela comunidade e ser de fato utilizada. Há abordagens que acabam por ficar nas monografias e dissertações ou ainda em artigos e livros, e não são de fato consumidas pelo público alvo.

Figura 30 – Questão 5 do Questionário de Avaliação

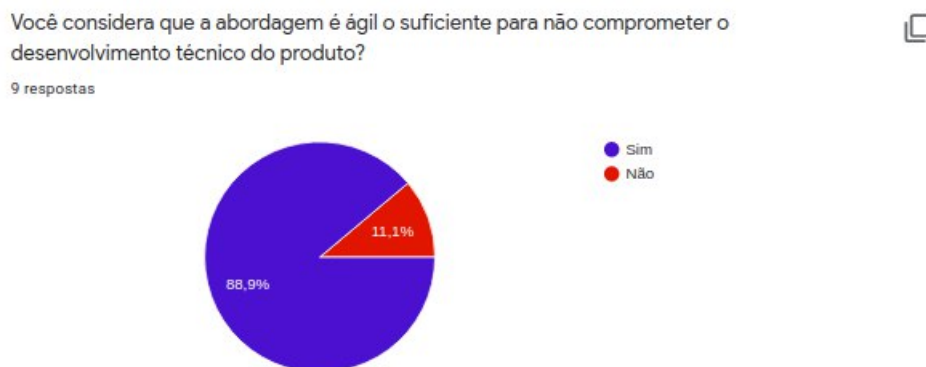


Fonte: Autor

Em seguida, conforme apresentado na Figura 31, a equipe é questionada quanto ao comprometimento do desenvolvimento técnico do produto mediante a aplicação da abordagem. Nesse ponto, a maior parte dos respondentes (88,9%) relata não identificar prejuízos na etapa técnica. Já 11,1% mencionou que sim, há algum prejuízo. No geral,

apontam como prejuízo o envolvimento em reuniões diárias para trabalhar com documentações ou realizar discussões negociais, no qual nem todos da equipe se identificam, com isso, reforça-se que diante das atividades propostas pela abordagem, sejam envolvidos apenas as pessoas chaves a depender do artefato que precisa ser produzido ou da pauta a ser discutida.

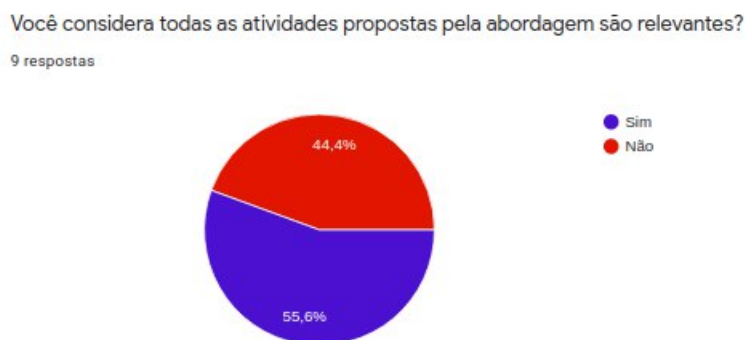
Figura 31 – Questão 6 do Questionário de Avaliação



Fonte: Autor

Na Figura 32, foi questionado o valor das atividades propostas durante o processo e parte da equipe considerou que nem todas as atividades propostas são realmente importantes para geração de valor ao produto, conforme havia sido levantado anteriormente mediante a aplicação da abordagem. Por exemplo, a própria sugestão de uma equipe com papéis bem definidos foi algo flexibilizado no Estudo de Caso realizado na empresa Vamos Parcelar.

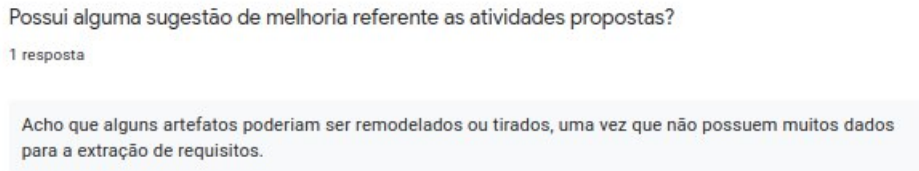
Figura 32 – Questão 7 do Questionário de Avaliação



Fonte: Autor

Na sequência, conforme apresentado na Figura 33, foi requisitado à equipe o levantamento de sugestões de melhorias referentes às atividades propostas durante a execução dos ciclos da abordagem. A sugestão baseou-se na adaptação dos artefatos, conforme realizado no próprio Estudo de Caso.

Figura 33 – Questão 8 do Questionário de Avaliação



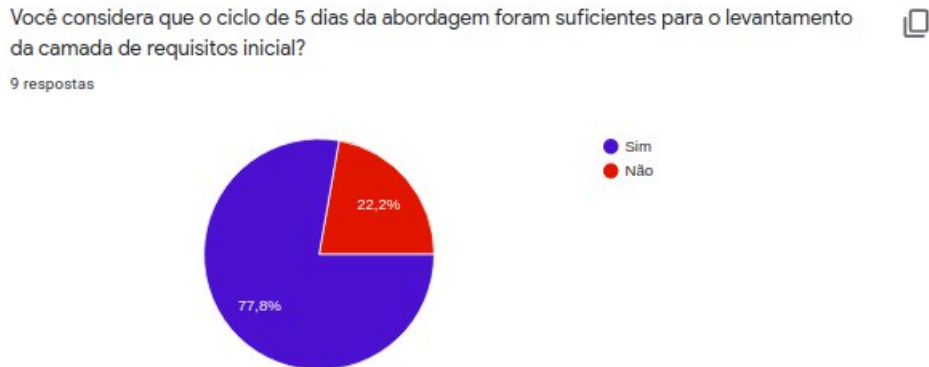
Fonte: Autor

Com relação ao prazo de execução da abordagem, definido como sendo cinco dias, foi questionado à equipe, conforme mostrado na Figura 34, se a mesma considera o tempo suficientemente hábil para abranger a etapa de requisitos inicial. Interessante notar que 77,8% responderam que sim, e 22,2% responderam que não. A pergunta foi ousada, pois mencionar que em 5 (cinco) dias tem como tratar essa etapa tão complexa, naturalmente, eram esperadas respostas predominantemente negativas. Mas, ao contrário, há predomínio de respostas positivas. Percebem-se dois fatores como conclusão: ou (i) as empresas estão tão orientadas às metodologias ágeis, e há tanto desconhecimento sobre a importância da Engenharia de Requisitos para a o sucesso de um software, que, para maioria, cinco dias atendem bem ao propósito, ou (ii) entende-se com as respostas que a maioria, na verdade, concorda que os cinco dias cobrem de forma satisfatória - e não completa - as demandas de requisitos. De toda forma, a *Requirements Sprint* é vista como sucesso.

Na Figura 35, é questionado à equipe se a mesma considera a abordagem flexível o suficiente para que seja adaptável dependendo da realidade do projeto em que está sendo inserida. A maior parte da equipe (89,9%) concordou que a mesma está apta à adaptações.

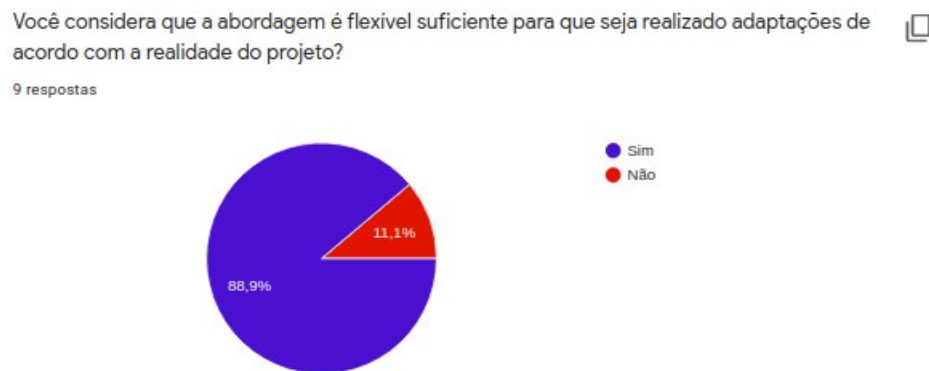
Como último questionamento, conforme apresentado na Figura 36, foi requisitado à equipe sobre o levantamento de sugestões de melhoria para a abordagem *Requirements Sprint*. A maior parte das sugestões foram baseadas na experiência de aplicação, conforme relatado no Estudo de Caso. Sendo assim, há necessidade, por exemplo, de adaptação e apresentação prévia da abordagem. A mesma ainda não pode ser oferecida sem um cuidado inicial de apresentação e moldagem às necessidades de cada organização. Tal processo é

Figura 34 – Questão 9 do Questionário de Avaliação



Fonte: Autor

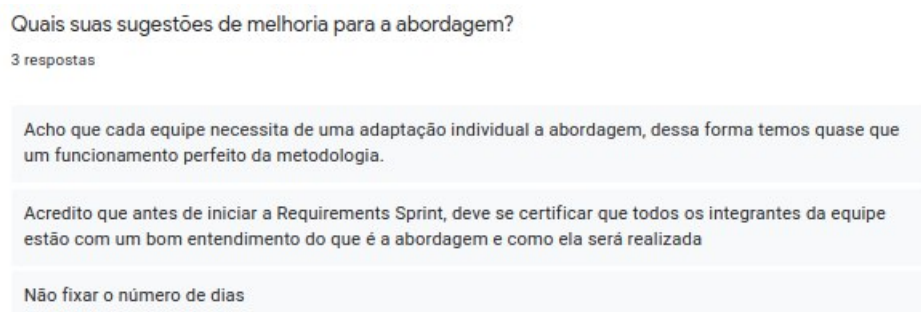
Figura 35 – Questão 10 do Questionário de Avaliação



Fonte: Autor

relativamente simples, conforme comentado no relato do Estudo de Caso, mas essencial para melhor condução da abordagem como um todo.

Figura 36 – Questão 11 do Questionário de Avaliação



Fonte: Autor

## 6.6 Resumo do Capítulo

Nesse capítulo, foi apresentado um Estudo de Caso, realizado em um projeto de iniciativa pública, de uma *fintech* voltada para o sistema financeiro. Na seção 6.4, foi realizada a especificação do processo de aplicação da abordagem no Estudo de Caso envolvendo a empresa Vamos Parcelar. Desta forma, com base em sua aplicação, foi possível comprovar sua eficácia com relação ao levantamento conciso de um escopo e requisitos que norteiam um projeto, de maneira flexível, conforme propõe a abordagem, e adaptável, a depender do contexto em que for inserida.

Mediante a inserção da abordagem, foi perceptível que deve haver um esforço com relação a conhecer bem a equipe para levantar quem são as pessoas chaves a serem envolvidas em cada uma das atividades, pois por mais relevante que seja ter a presença de desenvolvedores durante o processo, os mesmos não se sentem engajados em todas as atividades propostas, fazendo-se necessário mapear a presença dos mesmos para quando se tratar se pautas mais técnicas e nas demais atividades, permitir que os desenvolvedores possam utilizar o tempo para estudo das tecnologias. Dessa maneira centralizar um membro participante de todas as atividades para atualizar os desenvolvedores sobre os ganhos obtidos ao final de cada dia proposto.

Contudo, foi possível identificar um considerável nível de aceitação e compreensão da equipe com relação a sua aplicação e valor agregado, mediante ganhos referentes à etapa de requisitos, uma vez que a maioria dos envolvidos não tinha conhecimento estratégico de um processo para levantamento do escopo e requisitos sem que comprometesse o período de desenvolvimento e fosse realizada de maneira ágil e flexível suficiente, assim como ocorre no processo de desenvolvimento ágil do software, no qual existem processos que ocorrem de maneira paralela para evitar gargalos.

Como resultado, foi possível levantar e priorizar melhorias para a evolução da abordagem, assim como identificar algumas boas práticas executadas durante a aplicação da mesma, como por exemplo sua apresentação para a equipe e o refinamento das atividades e artefatos produzidos de acordo com a realidade do projeto, que mitigou grandes gargalos ao final do ciclo e na obtenção de resultados satisfatórios conforme o esperado. A abordagem refatorada, em sua nova versão, consta disponível em [Requirements Sprint](#).

## 7 Conclusão

Esse capítulo tem como objetivo apresentar uma visão geral de todos os resultados alcançados. Ele está dividido nas seguintes seções: [Considerações Finais](#), responsável evidenciar as contribuições desse trabalho, bem como o que foi feito para alcançar esses resultados, como o relato de um Estudo de Caso realizado para aprimoramento da abordagem a nível de procesos e ferramentas. A seção [Objetivos Realizados e Questão Respondida](#) retoma os objetivos desse trabalho bem como a questão de pesquisa estabelecida. Procura ainda conferir os status atuais quanto aos objetivos atingidos, bem como responder à questão de pesquisa. [Trabalhos Futuros](#) sugere possibilidades de continuação desse trabalho.

### 7.1 Considerações Finais

Esse trabalho teve como objetivo principal criar uma abordagem ágil e minimalista, para auxiliar as empresas e/ou organizações de pequeno e médio porte que se encontram em estágio inicial de amadurecimento a dar os primeiros passos na execução de atividades iniciais do processo de engenharia de requisitos. Para isso, foram sugeridos ciclos de execução de atividades que possuem como objetivo apoiar a equipe no levantamento de requisitos do projeto. Essa abordagem apoiou-se em modelos amplamente aceitos pela comunidade de Engenharia de Software, como por exemplo o *Design Sprint*. Para aperfeiçoar ainda mais a proposta das atividades sugeridas pela autora na abordagem, a mesma realizou um curso de facilitação de *Design Sprint* com o propósito de vivenciar experiências, de maneira mais próxima, com o método, e resgatar o que agregaria valor à metodologia *Requirements Sprint* para que a mesma pudesse ser incrementada.

Um dos principais fatores mencionados durante a aplicação do método *Design Sprint* é que o facilitador se assegure de que a empresa oferece um ambiente seguro para experimentação. Com isso, é necessário que a mesma esteja ciente da importância da implementação do método, como ocorreu no estudo de caso da *Requirements Sprint*, em que a abordagem foi apresentada e compreendida pelos membros de maneira anterior a sua aplicação para que fosse possível obter a contribuição dos membros com o máximo de qualidade e para que pudessem compreender o valor que seria gerado naquele determinado momento.

Com relação ao fator motivador de sua aplicação, assim como no *Design Sprint*,

a metodologia *Requirements Sprint* oferece como retorno financeiro para a empresa a mitigação de investimentos que não geram valor, devido ao fato de validar o escopo do projeto de maneira mais rápida. Como benefício, a empresa reduz custos desnecessários, obtém alinhamento técnico e comercial entre os membros, e acelerar o ciclo de inovação. Outro ponto mencionado com relação à execução do método *Design Sprint* é a atitude de fracionar os encontros com o especialista e/ou cliente, de forma que os encontros não fiquem espaçados, e não se perca o engajamento ou a lógica sequencial das discussões levantadas. Fundamentado neste ponto, justifica-se a proposta da abordagem de execução de 5 dias seguidos para aplicação das atividades.

Com relação às ferramentas, uma boa prática sugerida pelo método *Design Sprint* é a utilização de um *board*, como por exemplo a ferramenta *Miro*, para centralizar todo o fluxo de atividades propostas. Desta forma, o time consegue resgatar atividades executadas em ciclos anteriores e ter uma visão geral e de fácil acesso, permitindo assim que os mesmos possam criar blocos de anotações, puxar discussões com base nos registros, criar uma agenda, realizar dinâmicas para conhecer o grupo antes de iniciar a aplicação da metodologia, adicionar *timebox* direto pela ferramenta para gerenciar o tempo de execução das atividades, entre outras diversas funções da ferramenta que podem ser exploradas pelo time.

## 7.2 Objetivos Realizados e Questão Respondida

Para realização deste trabalho, foram considerados os seguintes objetivos específicos:

1. Definir, por meio de revisão sistemática, aspectos do método *Design Sprint* a serem utilizados no desenvolvimento da abordagem. *Status*: concluído;
2. Definir, por meio de revisão sistemática, aspectos do Ágil a serem utilizados no desenvolvimento da abordagem. *Status*: concluído;
3. Definir um conjunto de atividades, cujo embasamento é a literatura, de forma que a sua aplicação seja viável de ser realizada em uma *sprint* (*Requirements Sprint*), sendo essa dedicada às práticas da Engenharia de Requisitos e centrada em elicitação, modelagem e análise. *Status*: concluído;
4. Aplicar a abordagem, chamada *Requirements Sprint*, em um projeto ágil. *Status*: concluído, e
5. Avaliar os resultados coletados a partir da aplicação do processo resultante da abordagem. *Status*: concluído.



Tendo como objetivo guiar o trabalho, foi possível levantar e responder à seguinte questão de pesquisa: De que maneira a implementação conjunta de conceitos provenientes das práticas ágeis e do método *Design Sprint* pode contribuir para o desenvolvimento de uma abordagem capaz de apoiar a elicitação, a modelagem e a análise de requisitos em empresas e/ou organizações de pequeno ou médio portes?

A resposta à questão anterior pode ser conferida por meio de uma abordagem que combina Engenharia de Requisitos e Práticas Ágeis, conforme apresentado no capítulo *Requirements Sprint*

## 7.3 Trabalhos Futuros

Esse trabalho abre margem para vários trabalhos futuros. Dentre eles, podem ser mencionados:

- Refinamento de atividades sugeridas pela abordagem com base na aplicação de novas interações, para que seja possível acompanhar o andamento por exemplo do método *Design Sprint 2.0* que evolui a ponto de ser executado no prazo de 4 dias.
- Implantar a abordagem em empresas de grande porte para permitir avaliar a abordagem em vários contextos cognitivos, deixando mais claras as necessidades de evolução da abordagem para que sua aplicação possa abranger novos cenários.
- Disponibilizar as contribuições desse trabalho, para que seja evoluído colaborativamente, em parceria com outros interessados no tema. Essa evolução participativa permitirá, dentre outras aquisições, manter o catálogo sempre atualizado e alinhado às visões da comunidade especializada conforme evolução do tema.



## Referências

- ARIAS, M.; BUCCELLA, A.; CECHICH, A. A framework for managing requirements of software product lines. *Electronic Notes in Theoretical Computer Science*, Elsevier, v. 339, p. 5–20, 2018. Citado na página 33.
- ÁVILA, A. L.; SPÍNOLA, R. O. Introdução à engenharia de requisitos. *Engenharia de software Magazine*, p. 46–52, 2007. Citado 3 vezes nas páginas 33, 40 e 41.
- AWAD, M. A comparison between agile and traditional software development methodologies. *University of Western Australia*, p. 30, 2005. Citado 2 vezes nas páginas 42 e 43.
- BALDIN, N.; MUNHOZ, E. M. B. Snowball (bola de neve): uma técnica metodológica para pesquisa em educação ambiental comunitária. In: *CONGRESSO NACIONAL DE EDUCAÇÃO*. [S.l.: s.n.], 2011. v. 10, p. 329–341. Citado na página 28.
- BETTEREVALUATION. 2019. <<https://www.betterevaluation.org/en/evaluation-options/richpictures/>>. Acessado em 26/11/2019. Citado 2 vezes nas páginas 55 e 57.
- BIZAGI. 2019. <<https://www.bizagi.com/pt/produtos/bpm-suite/modeler>>. Acessado em 27/10/2019. Citado na página 60.
- CACCO. 2019. <<https://cacao.com/>>. Acessado em 24/11/2019. Citado 4 vezes nas páginas 54, 55, 56 e 58.
- CUEVAS, G.; SERRANO, A.; SERRANO, A. Assessment of the requirements management process using a two-stage questionnaire. In: IEEE. *Fourth International Conference on Quality Software, 2004. QSIC 2004. Proceedings*. [S.l.], 2004. p. 110–116. Citado 5 vezes nas páginas 33, 34, 141, 142 e 143.
- FAGAN, M. Reviews and inspections. *Software Pioneers—Contributions to Software Engineering*, p. 562–573, 2002. Citado na página 41.
- FIGMA. 2019. <<https://www.figma.com/>>. Acessado em 24/11/2019. Citado 2 vezes nas páginas 55 e 56.
- GAEA. 2019. <<https://gaea.com.br/o-que-voce-precisa-saber-sobre-metodologias-ageis-de-desenvolvimento/>>. Acessado em 04/11/2019. Citado na página 52.
- GERHARDT, T.; SILVEIRA, D. Métodos de pesquisa.[sl]: Plageder, 2009. 2019. Citado 2 vezes nas páginas 61 e 62.
- GIL, A. C. *Métodos e técnicas de pesquisa social*. [S.l.]: 6. ed. Editora Atlas SA, 2008. Citado 2 vezes nas páginas 62 e 63.

GIT. 2019. <<https://git-scm.com/about/>>. Acessado em 17/10/2019. Citado na página 49.

GITHUB. 2019. <<https://github.com/about/>>. Acessado em 17/10/2019. Citado na página 49.

JAMES, M.; WALTER, L. *Scrum Reference Card*. 2010–2018. Disponível em: <<http://scrumreferencecard.com/ScrumReferenceCard.pdf>>. Citado na página 34.

JUN, L.; QIUZHEN, W.; LIN, G. Application of agile requirement engineering in modest-sized information systems development. In: IEEE. *2010 Second world congress on software engineering*. [S.l.], 2010. v. 2, p. 207–210. Citado 2 vezes nas páginas 25 e 26.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. 2004, p. 1–26, 2004. Citado 2 vezes nas páginas 28 e 119.

KNAPP, J.; ZERATSKY, J.; KOWITZ, B. *Sprint: How to solve big problems and test new ideas in just five days*. [S.l.]: Simon and Schuster, 2016. Citado na página 26.

LARMAN, C. Applying uml and patterns: An introduction to object-oriented analysis and design and iterative development, third edition, addison wesley professional. *ISBN: 0-13-148906-2*, 2004. Citado na página 56.

LEE, S.; YONG, H.-S. Agile software development framework in a small project environment. *Journal of Information Processing Systems*, Korea Information Processing Society, v. 9, n. 1, p. 69–88, 2013. Citado 3 vezes nas páginas 44, 45 e 46.

LUCIDCHART. 2019. <[https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml#section\\_0](https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-sequencia-uml#section_0)>. Acessado em 04/11/2019. Citado na página 56.

MARTINS, G. A. Estudo de caso: uma reflexão sobre a aplicabilidade em pesquisas no brasil. *Revista de Contabilidade e Organizações*, Universidade de São Paulo, v. 2, n. 2, p. 8–18, 2008. Citado na página 63.

MOSCOW. 2019. <<https://www.productplan.com/glossary/moscow-prioritization/>>. Acessado em 28/11/2019. Citado 2 vezes nas páginas 81 e 132.

OLIVEIRA, M. F. D. Metodologia científica: um manual para a realização de pesquisas em administração. *Universidade Federal de Goiás. Catalão-GO*, 2011. Citado na página 61.

OVERLEAF. 2019. <<https://overleaf.com/>>. Acessado em 17/10/2019. Citado na página 51.

PRASAD, W. R. et al. Adopting design thinking practices to satisfy customer expectations in agile practices: A case from sri lankan software development industry. In: IEEE. *2018 Moratuwa Engineering Research Conference (MERCon)*. [S.l.], 2018. p. 471–476. Citado na página 26.

- RAFIQ USMAN E BAJWA, S. S. e. W. X. e. L. I. Técnicas de obtenção de requisitos aplicadas em startups de software. In: *2017 4<sup>ª</sup> Conferência Euromicro de Engenharia de Software e Aplicativos Avançados (SEAA)*. [S.l.: s.n.], 2017. p. 141–144. Citado 6 vezes nas páginas 25, 35, 36, 37, 38 e 39.
- RAUPP, F. M.; BEUREN, I. M. Metodologia da pesquisa aplicável às ciências. *Como elaborar trabalhos monográficos em contabilidade: teoria e prática*. São Paulo: Atlas, 2006. Citado na página 30.
- RIAZ, M. Q. et al. Customization of requirement engineering best practices for pakistan software industry. In: IEEE. *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. [S.l.], 2018. p. 1–6. Citado 3 vezes nas páginas 31, 32 e 42.
- SCRUM. 2019. <<https://www.scrum.org/>>. Acessado em 30/11/2019. Citado na página 44.
- SOMMERVILLE, I.; SAWYER, P. *Requirements engineering: a good practice guide*. [S.l.]: John Wiley & Sons, Inc., 1997. Citado na página 31.
- SPOLETINI, P.; FERRARI, A. Requirements elicitation: a look at the future through the lenses of the past. In: IEEE. *2017 IEEE 25th International Requirements Engineering Conference (RE)*. [S.l.], 2017. p. 476–477. Citado na página 27.
- SRIVASTAVA, A.; BHARDWAJ, S.; SARASWAT, S. Scrum model for agile methodology. In: IEEE. *2017 International Conference on Computing, Communication and Automation (ICCCA)*. [S.l.], 2017. p. 864–869. Citado 2 vezes nas páginas 42 e 43.
- TABASSAM SEHRISH E AL-QAHTANE, E. Estudo comparativo das técnicas de modelagem de engenharia de requisitos do e-guide pessoal do al-haj. In: *2019 2ª Conferência Internacional sobre Aplicações Informáticas e Segurança da Informação (ICCAIS)*. [S.l.: s.n.], 2019. p. 1–8. Citado na página 25.
- TRELLO. 2019. <<https://trello.com/>>. Acessado em 17/10/2019. Citado 3 vezes nas páginas 50, 57 e 58.
- TURINE, M. A. S.; MASIERO, P. C. Especificação de requisitos: uma introdução. *Relatório Técnico. ICMSC: USP*, 1996. Citado na página 40.
- VELHOBIT. 2019. <<https://velhobit.com.br/design/wireframe-mockup-ou-prototipo.html>>. Acessado em 04/11/2019. Citado na página 54.
- VLAANDEREN, K. et al. The agile requirements refinery: Applying scrum principles to software product management. *Information and software technology*, Elsevier, v. 53, n. 1, p. 58–70, 2011. Citado na página 42.
- WOHLIN, C. et al. *Engineering and managing software requirements*. [S.l.]: Springer Science & Business Media, 2005. Citado na página 35.
- YIN, R. K. Estudo de caso: planejamento e métodos. 2a edição. *Porto Alegre*, 2001. Citado 3 vezes nas páginas 63, 64 e 88.

ZENHUB. 2019. <<https://www.zenhub.com/>>. Acessado em 09/11/2019. Citado na página 59.

ZHANG, X. S.; DORN, B. Accelerating software development through agile practices-a case study of a small-scale, time-intensive web development project at a college-level it competition. *Journal of Information Technology Education*, v. 11, p. 25–37, 2012. Citado na página 44.

ZOTERO. 2019. <<https://www.zotero.org/>>. Acessado em 17/10/2019. Citado na página 52.

# Apêndices





# APÊNDICE A – Revisão Sistemática

Para a concepção e definição de um protocolo de revisão que visa, especificar a pesquisa, as questões e os métodos utilizados, foi produzido neste apêndice um resumo das etapas de uma revisão sistemática definidas em três fases principais por (KITCHENHAM, 2004) que serão aplicadas na execução deste estudo secundário:

## A.1 Planejamento da Revisão:

As etapas associadas ao planejamento da revisão são:

### A.1.1 Desenvolvimento de um Protocolo de Revisão.

Foi desenvolvido um protocolo de revisão, o qual possibilita, a outros pesquisadores, a repetição da pesquisa. O mesmo é denotado abaixo.

#### A.1.1.1 Questões de Pesquisa

Para definição das questões utilizadas para pesquisa foi utilizado como critério a relação com a tema deliberado preliminarmente. Dessa forma foi acordado a existência de perguntas de acordo com a base utilizada apenas nos idiomas:

- Português
  - Inglês
- 
- Existem trabalhos que correlatam as práticas das equipes ágeis no processo de elicitação. modelagem e análise de requisitos?
  - O que é feito nos níveis mais altos de abstração?
  - Quanto tempo dedicam à etapa de levantamento de requisitos de software?
  - Quais autores acordam boas práticas para um bom processo de elicitação, modelagem e análise de requisitos?
  - De que maneira o método Design Sprint reduz o processo de design de um projeto para 5 dias?
  - Agile practices in requirements management

- Elicitation process, modeling and requirements analysis
- Scope Management Process
- Method of the Sprint design process

#### A.1.1.2 Definição das Bases de Pesquisa

Como bases de pesquisa, foram definidas as seguintes:

- CAPES
- IEEE
- WEB OF SCIENCE
- ACM
- SCOPUS

#### A.1.1.3 Palavras-Chave

Como etapa do processo de seleção de artigos, foi utilizado como método de seleção a busca pelas seguintes palavras-chave:

- Sprint Design
- Elicitation / Elicitação
- Requirements / Requisitos
- Agile practices / Práticas Ágeis
- Modeling / Modelagem
- Analysis / Análise
- Startups
- Modeling Techniques / Técnicas de Modelagem
- Project Performance / Desempenho do projeto
- Small scale projects / Projetos de pequena escala

#### A.1.1.4 Procedimento de Seleção de Artigo

Para realizar a seleção dos artigos a serem estudados, foram definidos critérios de inclusão e exclusão de artigos, sendo necessário para evitar que artigos indesejados sejam incluídos na pesquisa.

#### A.1.1.5 Avaliação da Qualidade

A avaliação da qualidade do artigo deu-se por meio da análise de seu conteúdo, com foco principalmente no capítulo de introdução, resultados e conclusões. Sendo que para isso foram definidas perguntas a serem respondidas para validação da qualidade do artigo:

1. O estudo apresentado é condizente com os objetivos da pesquisa?
2. As evidências apresentadas são relevantes para o desenvolvimento da abordagem?
3. O artigo apresenta qualis igual ou superior a B1?
4. O artigo se encontra em uma das categorias (dissertações, teses, conferências ou *journal*?)

#### A.1.1.6 Processo de Análise de Artigo

1. Busca na Base
2. Leitura do título com base nos critérios de seleção
3. Avaliação da disponibilidade de leitura do artigo
4. Leitura do resumo com base nos interesses pertinentes ao tema
5. Avaliação da qualidade do artigo com base na Avaliação da Qualidade
6. Leitura do artigo completo

#### A.1.1.7 Processo de Inclusão e Exclusão

Inclusão:

- Artigos relacionados ao processo de gerenciamento de requisitos
- Artigos relacionados a aplicação de práticas ágeis
- Artigos publicados a partir de 2004

Exclusão:

- Artigos indisponíveis para leitura
- Artigos que possuem outro idioma que não seja Português ou Inglês
- Artigos não relacionados a área de tecnologia



# APÊNDICE B – Abordagem Preliminar

Neste capítulo, é apresentada a abordagem preliminar, que por sua vez orienta-se por atividades voltadas para o processo de engenharia de requisitos e conceitos advindos do método *Design Sprint* e da metodologia Ágil. Em um primeiro momento, tem-se a [Proposta Preliminar](#), onde serão apresentados os princípios e objetivos da aplicação da abordagem. Em seguida, é apresentada uma sugestão de [Composição da Equipe](#) para guiar de forma mais estruturada a execução das atividades, bem como é acordada a [Estruturação das Atividades](#) para maior detalhamento com relação ao objetivo de entrega de cada referido dia e sugestão de atividades para viabilizar este objetivo. Por fim, tem-se o [Resumo do Capítulo](#).

## B.1 Proposta Preliminar

Conforme indicado no [capítulo 1](#), esse trabalho de conclusão de curso consistiu na elaboração de uma abordagem ágil e minimalista, a qual possui como intuito principal auxiliar as empresas e/ou organizações de pequeno e médio porte que se encontram em estágio inicial de amadurecimento a dar os primeiros passos na execução de atividades iniciais do processo de engenharia de requisitos.

Com relação à entrega referente à primeira etapa do trabalho, uma versão preliminar da abordagem foi proposta. Para o desenvolvimento desta versão preliminar, foram esboçadas algumas ideias e realizados levantamentos do que de fato agregaria valor quando em termos de execução de um processo que envolve atividades de engenharia de requisitos e metodologia ágil, inseridos ainda em um contexto de curto prazo como o de desenvolvimento do método *Design Sprint*. Dentre os elementos agregadores de valor nesse sentido, têm-se aspectos que a equipe precisa lidar, aceitar e executar .

### B.1.1 Esboço da Abordagem Preliminar

A partir deste mapeamento, foram esboçadas ideias iniciais conforme apresentado na Figura 23

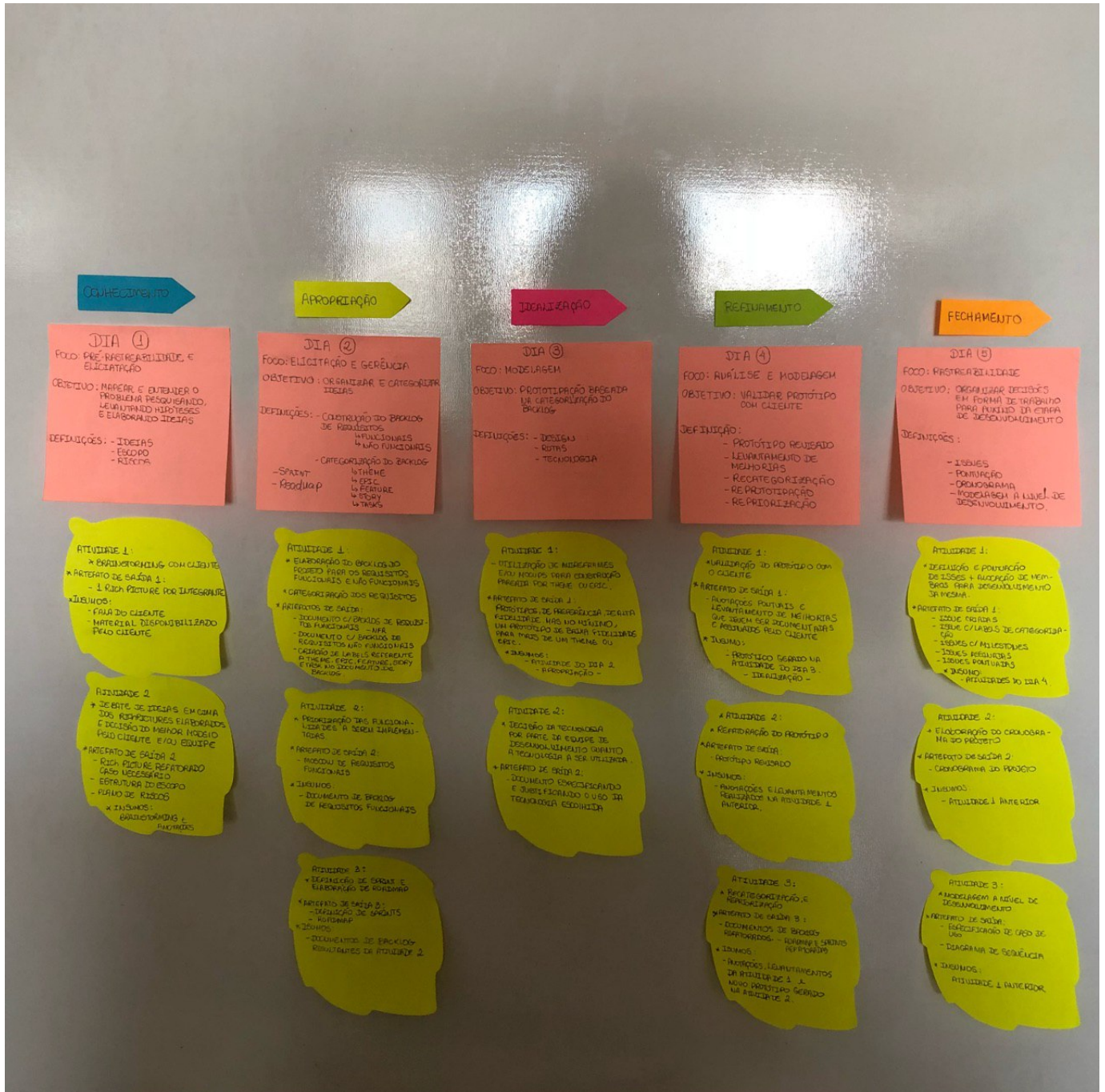


Figura 37 – Esboço da abordagem preliminar

Fonte: Autor

## B.1.2 Questionamentos

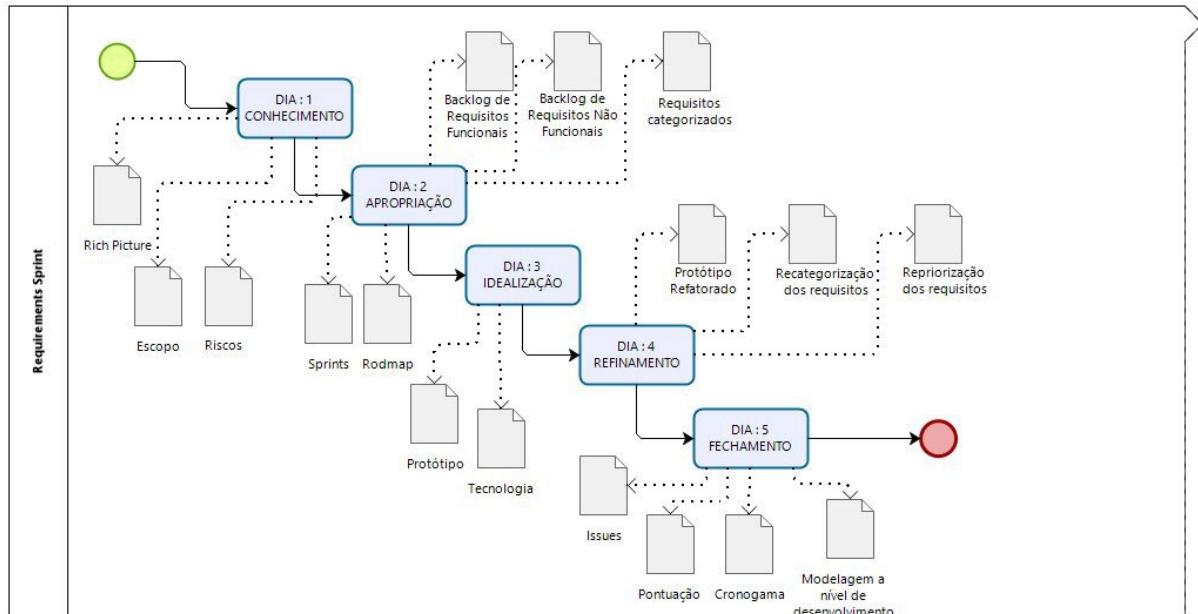
Foram levantados alguns questionamentos quanto às atividades propostas para o desenvolvimento da abordagem, os quais são listados a seguir:

- Quais artefatos de saída mapeados no processo a equipe pode negligenciar por não achar fundamental para o processo? Como lidar com essa questão?
  - O fato da equipe negligenciar alguns artefatos de saída propostos pela abordagem é um risco que se assume. Porém, a ideia da abordagem é sugerir atividades de forma que seja o mais flexível possível ao ponto da equipe conseguir definir se se encaixa ou não em seu perfil ou de substituir por algum artefato que gere mais valor ao produto em questão.
- Quais artefatos podem ser gerados apenas em situações críticas de desenvolvimento visando dosar o trabalho da equipe? Como identificar que a atividade é mapeada como crítica para o projeto a ponto de necessitar de uma melhor documentação?
  - Alguns artefatos que demandam maior tempo e dedicação da equipe, e que não agregam tanto valor ao produto, mas sim em relação à rastreabilidade e à memória técnica, serão produzidos apenas em situações críticas, as quais serão definidas pela própria equipe. Sugere-se o uso de métricas, aplicadas e coletadas ao longo do processo de desenvolvimento, visando estabelecer se uma dada situação é ou não crítica.

## B.1.3 Desenho da Abordagem Preliminar

A Figura 24 apresenta a abordagem preliminar desenhada por meio da ferramenta *Bizagi*, na qual é mapeado inicialmente o fluxo de desenvolvimento da mesma, bem como a sugestão de artefatos gerados em cada um dos dias inseridos na *sprint* dedicada à aplicação da abordagem *Requirements Sprint*. Cabe ressaltar que serão gerados alguns *templates* visando auxiliar ainda mais as equipes ágeis na elaboração dos artefatos sugeridos. Por exemplo, *templates* para o plano de riscos e definição de escopo, da atividade DIA1: Conhecimento, ou ainda *template* para elaboração do Roadmap, da atividade DIA 2: Apropriação. Os *templates* constarão como Apêndices na monografia do TCC 02. Outros suportes encontram-se já apresentados no capítulo 3, tais como: *Rich Picture* e Protótipo.

Figura 38 – Fluxo da abordagem preliminar proposta



Powered by  
**bizagi**  
Modeler

Fonte: Autor

#### B.1.4 Princípios para Aplicação da Abordagem

Um dos principais objetivos das atividades presentes na abordagem proposta é que possam ser simples, de modo que os objetivos da execução de cada um dos dias propostos tenham sido concluídos ao final da *sprint*.

Durante a criação das atividades, embora primariamente guiadas as necessidades gerais de composição de produto, foi levado em conta cada princípio aplicável do *Scrum*. A seguir, são apresentados os princípios, junto a identificadores, para que se possa manter uma relação entre estes e as atividades sugeridas.

- **P1 - Empirismo**

Aprendendo com a experiência. Após rodar algumas *sprints* dedicadas à aplicação da *Requirements Sprint* (da maneira correta, executando cada dia proposto pela abordagem), você começará a gerar o seu próprio conhecimento empírico, realimentado a abordagem e aumentado a qualidade e a definição dos melhores artefatos a serem gerados, e que realmente contribuem com o perfil da equipe e do projeto.

- **P2 - Auto-organização**



Quanto maior a alocação de esforço com relação à auto-organização da equipe referente à aplicação das atividades sugeridas pela abordagem, maiores serão os resultados alcançados. Quanto mais auto-organizada for a equipe, maior será a entrega de valor, uma vez que a abordagem conta com uma boa dedicação de entregas e decisões que devem ser concluídas ao final de cada dia.

- **P3 - Colaboração**

Dedica-se as três dimensões básicas referentes ao trabalho colaborativo: consciência, articulação e apropriação, de forma a defender o gerenciamento de projetos como um processo de criação de valor compartilhado, com todas as frentes trabalhando e interagindo em conjunto para atingirem os melhores resultados possíveis por meio da combinação de diferentes visões de produto.

- **P4 - Priorização baseada em Valor**

Neste princípio, destaca-se o máximo de entrega de valor de negócio possível, durante toda execução do projeto. Entregar valor é atender a demanda do seu mercado e/ou resolver o problema do seu cliente, sabendo priorizar da forma correta o que se entende como o sucesso na execução desta atividade.

- **P5 - *Time-boxing***

Os elementos *Time-boxed* referentes à metodologia Scrum incluem as *Sprints*, as Reuniões Diárias, a Reunião de Planejamento da *Sprint*, e a Reunião de Revisão da *Sprint*. Este conceito de *Time-boxed* é aplicado na abordagem para que a equipe alcance a produtividade e consiga ainda melhorar seu comportamento com relação ao princípio anteriormente mencionado, referente à auto-organização, ou seja, neste contexto, um princípio contribui com o outro. O princípio de realização de atividades realizadas com prazo e duração limitadas norteia a equipe para que os objetivos levantados sejam atingidos de maneira eficaz e eficiente.

## B.2 Composição da Equipe

Com relação à composição da equipe para realização da aplicação da abordagem no contexto do projeto por meio da dedicação da equipe pelo período de uma *sprint*, não existe uma exigência para que se obtenha sucesso na inserção da mesma. A abordagem faz uso de sugestões a fim de orientar a equipe quanto a participações relevantes de acordo com o contexto da empresa e sua composição comum.

Logo, não existe uma exigência de papéis específicos que devem estar presentes, apenas sugestões para que a empresa possa se apropriar da composição que melhor lhe

favorece e que esteja inserida em sua realidade de trabalho atual. Levando em consideração o fato de que a abordagem foi desenvolvida em conjunto com os conceitos advindos do método *Design Sprint*, os seguintes membros são sugeridos para composição da equipe:

- *Designer* (Pessoa responsável por tratar a parte referente ao design do que se pretende desenvolver);
- *Product Manager/Product Owner* (Pessoa que conhece a fundo todo produto e seus desafios);
- *Stakeholder* (Parte interessada no negócio);
- Desenvolvedores ou membro com conhecimento técnico (Responsáveis por transformar as ideias advindas do processo inicial de concepção em código), e
- Mediador ou facilitador (Responsável por guiar a equipe para que não se perca o foco, sendo também o responsável por conduzir as sessões coletivas).

### B.3 Estruturação das Atividades

Além do mapeamento inicial realizado anteriormente, cada atividade possui **Dia**; **Tema**; **Foco**; **Objetivo**; **Definições**; **Atividades**; com seus respectivos **Insumos** e **Artefatos de Saída** sugeridos, assim como o **Princípio do Scrum** trabalhado.

- **Dia**: Possui como objetivo especificar de forma numérica e crescente a que dia se refere determinada atividade;
- **Tema**: Possui como objetivo especificar o tipo do conjunto de atividades que será realizada no determinado dia;
- **Foco**: Possui como objetivo especificar quais atividades de requisitos são tratadas no determinado dia;
- **Objetivo**: Utilizado para especificar o que se pretende alcançar com a execução de atividades propostas no referido dia;
- **Definições**: Possui como objetivo determinar o que ao final do dia se deseja ter definido;
- **Atividades**: Possui como objetivo especificar quais as atividades sugeridas pela abordagem serão realizadas pela equipe para alcance de suas metas;

- **Insumos:** Possui como objetivo especificar o material que servirá de apoio para execução das atividades pela equipe;
- **Artefato de Saída:** Possui como objetivo especificar quais serão os documentos gerados ao final da aplicação de cada atividade proposta;
- **Princípio do *Scrum*:** Possui como objetivo especificar qual princípio do scrum está sendo utilizado nas atividades do determinado dia.

### B.3.1 Descrição da Programação

A programação referente à *sprint* é descrita a seguir, utilizando a estrutura acordada anteriormente por meio da utilização de uma linguagem informal de maneira proposital com o objetivo de aproximar o leitor.

**Dia:** 1

**Tema:** CONHECIMENTO

**Foco:** Pré-Rastreabilidade e Elicitação

**Objetivo:** Mapear e entender o problema pesquisando, levantando hipóteses e elaborando ideias.

**Definições:**

- Ideias
- Escopo
- Riscos

**Atividades:**

Estão programadas 2 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados.

**Atividade 1:**

### *Brainstorming* com o cliente/*stakeholder*

- Insumos:
  - Fala do cliente
  - Material disponibilizado pelo cliente
- Artefatos de Saída:
  - 1 *Rich Picture* por membro da equipe
- Princípio(s) *Scrum*: P2, P3 e P5

### **Atividade 2:**

Debate de ideias com base nos *Rich Pictures* elaborados e tomada de decisão do melhor modelo a ser seguido pelo cliente e/ou equipe.

- Insumos:
    - *Brainstorming* com cliente
    - Anotações
  - Artefatos de Saída:
    - *Rich Picture* refatorado, caso necessário
    - Definição do Escopo
    - Plano de Riscos
  - Princípio(s) *Scrum*: P1, P2, P3 e P5
- 

**Dia:** 2

**Tema:** APROPRIAÇÃO

**Foco:** Elicitação e Categorização

**Objetivo:** Organizar e categorizar ideias.

**Definições:**

- Construção do *backlog* de requisitos: Funcionais e Não Funcionais

- Categorização do *backlog* em: *Theme, Epic, Feature, Story e Task*
- *Sprints*
- *RoadMap*

### **Atividades:**

Estão programadas 3 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados.

#### **Atividade 1:**

Elaboração do *backlog* do projeto para os requisitos funcionais e não funcionais.

- Insumos:
  - Escopo definido no dia anterior
- Artefatos de Saída:
  - Documento com definição de *backlog* de requisitos funcionais
  - Documento com definição de *backlog* de requisitos não funcionais
  - Criação de labels referentes a *theme, epic, feature, story e task* inseridas ao documento de *backlog*.
- Princípio(s) *Scrum*: P1, P2, P3 e P5

#### **Atividade 2:**

Priorização das funcionalidades a serem implementadas.

- Insumos:
  - Documentos referente ao *backlog* de requisitos funcionais gerado na atividade anterior

- Artefatos de Saída:
  - Backlog priorizado.
- Princípio(s) *Scrum*: P1,P2, P3, P4 e P5

### **Atividade 3:**

Definição de *sprints* e elaboração de *roadmap*.

- Insumos:
  - Documentos de *backlog* e priorização gerados nas atividades anteriores.
- Artefatos de Saída:
  - Definição de *sprints*
  - Roadmap
- Princípio(s) *Scrum*: P1,P2, P3, P4 e P5

### **Observações:**

Sugere-se para priorização do backlog o uso da técnica de priorização *Moscow* ([MOSCOW, 2019](#)), por ser simples e permitir priorizar em quatro níveis de granularidade, sendo: *Must have*, *Should have*, *Could have* e *Won't have*. Mas, caso a equipe já faça uso de outro recurso para conduzir a priorização, a abordagem é flexível nesse ponto.

.

---

**Dia:** 3

**Tema:** IDEALIZAÇÃO

**Foco:** Modelagem

**Objetivo:** Prototipação baseada na categorização do *backlog*.

**Definições:**

- *Design*
- Rotas

- Tecnologias

**Atividades:**

Estão programadas 2 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados e a priorização baseada em valor para se obter sucesso na execução de uma atividade mapeada para este dia.

**Atividade 1:**

Utilização de *wireframes* e/ou *mockups* para construção pareada por *theme* ou *epic*.

- Insumos:
  - Atividade do dia 2 - APROPRIAÇÃO -
- Artefatos de Saída:
  - Protótipos de preferência de alta fidelidade, mas no mínimo um protótipo de baixa fidelidade para mais de um *theme* ou *epic*.
- Princípio(s) *Scrum*: P1,P2, P3 e P5

**Atividade 2:**

Decisão da(s) tecnologia(s) por parte da equipe de desenvolvimento quanto a que será aplicada ao projeto.

- Insumos:
  - Protótipo gerado na atividade anterior
- Artefatos de Saída:
  - Documento especificando e justificando uso da tecnologia selecionada pela equipe.
- Princípio(s) *Scrum*: P1,P2, P3 e P5

**Dia:** 4

**Tema:** REFINAMENTO

**Foco:** Modelagem e Análise

**Objetivo:** Validar protótipo com cliente e alinhar o que for necessário para se obter a ideia mais concreta possível do produto com relação à visão e às necessidades do cliente.

**Definições:**

- Protótipo revisado
- Levantamento de melhorias
- Recategorização
- Reprototipação
- Repriorização

**Atividades:**

Estão programadas 3 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados e a priorização baseada em valor para se obter sucesso na execução de uma atividade mapeada para este dia.

**Atividade 1:**

Validação do protótipo com o cliente

- Insumos:
  - Protótipo gerado na atividade do dia 3 - IDEALIZAÇÃO -
- Artefatos de Saída:



- Anotações pontuais e levantamento de melhorias que devem ser documentadas e de preferência assinadas pelo cliente.

- Princípio(s) *Scrum*: P1,P2, P3 e P5

### **Atividade 2:**

Refatoração do protótipo, caso seja necessário.

- Insumos:
  - Anotações e levantamentos realizados na atividade anterior.
- Artefatos de Saída:
  - Protótipo revisado e refatorado
- Princípio(s) *Scrum*: P1,P2, P3 e P5

### **Atividade 3:**

Recategorização e Repriorização.

- Insumos:
    - Anotações, levantamentos realizados na atividade 1 e novo protótipo gerado na atividade 2.
  - Artefatos de Saída:
    - Documentos de *backlog* refatorados
    - *Sprints* refatoradas
    - *Roadmap* refatorados
  - Princípio(s) *Scrum*: P1,P2, P3, P4 e P5
- 

**Dia:** 5

**Tema:** FECHAMENTO

**Foco:** Rastreabilidade

**Objetivo:** Organizar decisões, visando estabelecer uma rotina de trabalho com tarefas que facilitem o desenvolvimento ágil do produto desejado.

**Definições:**

- *Issues*
- Pontuação
- Cronograma
- Modelagem a nível de desenvolvimento

**Atividades:**

Estão programadas 3 atividades para este dia, de forma que se torna necessário fazer uso de um *time-boxed* para que a realização das atividades sejam concluídas dentro de um prazo determinado pela equipe por meio do princípio de Auto-organização. A ideia é garantir a execução das atividades, assim como estimular o princípio de Empirismo por parte da equipe. Esse último princípio permitirá à equipe definir com mais clareza e precisão, com base nas experiências adquiridas, o que de fato agrega valor ao produto final desejado em termos de artefatos gerados e a priorização baseada em valor para se obter sucesso na execução de uma atividade mapeada para este dia.

**Atividade 1:**

Definição e pontuação de *issues*, bem como a alocação de membros para desenvolvimento da mesma.

- Insumos:
  - Atividades realizadas referentes ao dia 4
- Artefatos de Saída:
  - *Issues* criadas
  - *Issues* com *labels* de categorização
  - *Issues* com *milestones*
  - *Issues* assinadas
  - *Issues* pontuadas
- Princípio(s) *Scrum*: P1, P2, P3, P4 e P5

**Atividade 2:**

Elaboração do cronograma do projeto

- Insumos:
  - Atividade 1 anterior
- Artefatos de Saída:
  - Cronograma do projeto
- Princípio(s) *Scrum*: P1,P2, P3, P4 e P5

**Atividade 3:**

Modelagem a nível de desenvolvimento.

- Insumos:
  - Atividade 1 anterior.
- Artefatos de Saída:
  - Diagrama de sequência
- Princípio(s) *Scrum*: P1,P2, P3 e P5

**Observações:** Atividade sugerida para ser aplicada de forma opcional pela equipe.



# Anexos



# ANEXO A – Avaliação do processo de gestão de requisitos utilizando um questionário de duas fases

Os autores (CUEVAS; SERRANO; SERRANO, 2004) propõe uma metodologia de avaliação baseada em um questionário de dois estágios para identificar quais práticas do processo de gestão de requisitos são executadas, mas não documentadas; quais práticas exigem ser priorizadas, e quais não são implementadas devido à má gestão ou desconhecimento. Dessa forma, a fim de validar a metodologia de avaliação, o questionário foi aplicado por esses autores em um estudo.

Ressalta-se que o questionário proposto pelos autores baseou-se nas práticas dos processos de gestão de requisitos estabelecidos pelo CMMI (Modelo Integrado de Maturidade em Capacitação), sendo dividido em duas etapas:

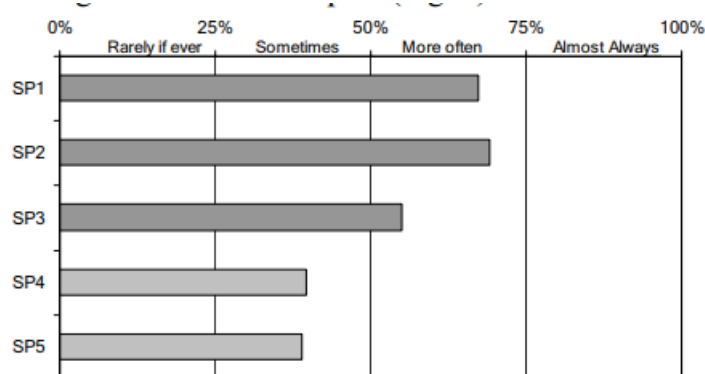
- A primeira etapa definida por (CUEVAS; SERRANO; SERRANO, 2004) relaciona-se com as práticas específicas, sendo aplicada aos funcionários que executam os processos como: analista, desenvolvedores, *designers*, programadores, entre outros. Esta fase é dividida em cinco práticas específicas conforme apresentadas no quadro X.
- Já a segunda etapa, de acordo com (CUEVAS; SERRANO; SERRANO, 2004), é aplicada aos funcionários que gerenciam o processo como: *senior*, gerente de software ou líderes de equipe. Esta etapa é dividida em dez práticas genéricas conforme apresentadas no quadro Y. Pretende-se por meio desta identificar se o gerenciamento de requisitos está ou não institucionalizado.

A análise dos valores obtidos a partir das respostas dadas por meio da aplicação da primeira etapa do questionário a este estudo de casos constatou, conforme mencionado pelos autores, que nenhuma das cinco práticas específicas atinge o nível de desempenho mínimo que foi definido como (75%) para ser considerado como ponto forte no processo. No entanto, existem três práticas, SP1, SP2 e SP3, que ficaram entre 50% e 75% no nível de percentual do desempenho, sugerindo que o esforço de melhoria poderia ser focado apenas para documentar o processo.

Por outro lado, os valores obtidos para as práticas SP4 e SP5 estavam em 50%. Este resultado sugere que essas práticas devem ser priorizadas em um plano de ação organizacional. Com base na análise dos valores obtidos a partir das respostas dadas na aplicação da segunda etapa do questionário a este estudo de casos, foi constatado que nenhuma das dez práticas genéricas atinge o nível mínimo de desempenho 75% para que seja considerado um processo institucionalizado.

Entretanto, em (CUEVAS; SERRANO; SERRANO, 2004) afirma-se que esta observação foi esperada, uma vez que nenhuma Prática Específica foi classificada igual a 75%. O maior valor obtido a partir do segundo estágio do questionário foi "GP3-fornecer o adequado recurso para executar o processo de RM (Modelagem de Requisito)". Isto significa que esta Prática Geral é realizada, mas apenas algumas vezes é documentada. Para este caso, o plano de ação deve ser focado para documentação da prática. Estes resultados são exibidos nas Figuras 23 e 24 a seguir apresentadas.

Figura 39 – Resultado das práticas específicas aplicadas.



**Fig. 1. RM specific practices perform level**

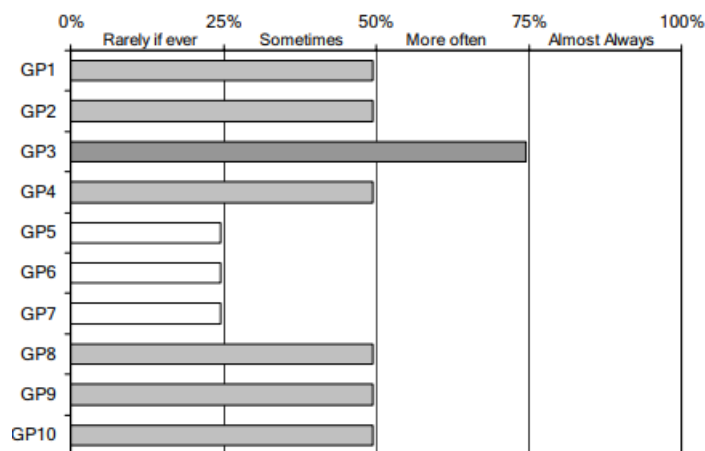
Fonte: (CUEVAS; SERRANO; SERRANO, 2004)

Como conclusão, os autores (CUEVAS; SERRANO; SERRANO, 2004) apontam que a metodologia de avaliação alternativa, baseada em duas etapas do questionário, poderia fornecer informações valiosas relacionadas às áreas que exigem priorização. Sendo assim, nesse estudo de casos, duas práticas específicas e três práticas genéricas mostraram alguns problemas. Tal colocação sugere que sejam priorizados esses aspectos no plano de ação.

Outra vantagem acordada pelos autores quanto à metodologia é o auxílio na redução de custo, tempo e esforço de avaliação. Um projeto típico requer uma média de 28 dias para o processo de avaliação e para derivar alguns resultados. No estudo de casos,



Figura 40 – Resultado das práticas genéricas aplicadas.



**Fig. 7. Generic practices perform level**

Fonte: (CUEVAS; SERRANO; SERRANO, 2004)

o uso da metodologia ajudou a reduzir a avaliação a apenas dez dias para apresentar os resultados e o plano de ação.

Por fim, consta ainda no artigo dos autores a afirmação de que a avaliação não fornece qualquer melhoria por si só, mas levanta informações valiosas sobre o estado atual do processo, proporcionando a base para melhores escolhas sobre as mudanças que os profissionais da tecnologia da informação devem fazer.

Como resultado da avaliação, deve ser gerado um plano de ação, a fim de prosseguir com o projeto de melhoria. O plano de ação descreve todas as atividades, entregáveis, cronograma e priorização de processos a serem melhorados.