

VINICIUS PRODOCIMO

DESIGN THINKING NA ENGENHARIA DE  
REQUISITOS EM PROJETOS DE SOFTWARE:  
ESTUDOS DE CASO NO BRASIL

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná para obtenção do título de Mestre em Informática.

Curitiba  
2020

VINICIUS PRODOCIMO

DESIGN THINKING NA ENGENHARIA DE  
REQUISITOS EM PROJETOS DE SOFTWARE:  
ESTUDOS DE CASO NO BRASIL

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná para obtenção do título de Mestre em Informática.

Área de concentração: Ciência da Computação

Orientadora: Prof.<sup>a</sup> Dra. Andreia Malucelli  
Coorientadora: Prof.<sup>a</sup> Dra. Sheila Reinehr

Curitiba  
2020

Dados da Catalogação na Publicação  
Pontifícia Universidade Católica do Paraná  
Sistema Integrado de Bibliotecas – SIBI/PUCPR  
Biblioteca Central  
Edilene de Oliveira dos Santos CRB-9/1636

P964d 2020	Prodocimo, Vinicius Design thinking na engenharia de requisitos de software : estudos de caso no Brasil / Vinicius Prodocimo ; orientadora, Andreia Malucelli ; coorientadora, Sheila Reinehr. -- 2020 112 f. : il. ; 30 cm  Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2020. Bibliografia: f. 104-108  1. Engenharia de software. 2. Software – Desenvolvimento. 3. Design thinking. 4. Projetos de engenharia. I. Malucelli, Andreia. II. Reinehr, Sheila dos Santos. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título.  CDD. 20.ed. – 005.1
---------------	--



**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
ESCOLA POLITECNICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**AUTORIZAÇÃO PARA ENTREGA DA  
VERSÃO FINAL DA  
DISSERTAÇÃO DE MESTRADO**

Eu, Prof.<sup>a</sup> Dra. Andreia Malucelli, orientadora do aluno de mestrado Vinicius Prodociro, certifico que foram feitas todas as alterações solicitadas pela Banca Examinadora e indicadas na ata de defesa, e que a dissertação intitulada DESIGN THINKING NA ENGENHARIA DE REQUISITOS EM PROJETOS DE SOFTWARE: ESTUDOS DE CASO NO BRASIL, na presente forma, cumpre todas as normas de formatação definidas pelo Programa, podendo, portanto, ser entregue na secretaria do Programa.

Curitiba (PR), 09 de setembro de 2020.

Orientadora

## DEDICATÓRIAS

Para minha esposa Maritana e minha filha Melissa que sempre estiveram ao meu lado durante os desafios e conquistas desta pesquisa.

À minha família por me propiciar a educação necessária para chegar até este ponto.

## **AGRADECIMENTOS**

Às minhas orientadoras Andreia Malucelli e Sheila Reinehr pela compreensão, cooperação, ensinamentos e paciência em me guiar por este caminho e por acreditarem em mim.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001.

*“O que vale na vida não é o ponto de partida e sim a caminhada. Caminhando e semeando, no fim terás o que colher.”*

*Cora Coralina*

## ABSTRACT

Organizations, driven by business digitization, have in software the main core of value generation and the main channel of communication with their clients. The software, as well as responding to momentary market needs, spans an extensive product family, ranging from mobile applications to multilateral platforms. Thus, the software specification needs to represent solutions focused on consumer problems and market needs. However, Requirements Engineering, whose approach is strongly linked to technology, becomes deficient and ineffective when the problem is not well defined or when looking for an innovative solution, thus needing a complementary approach. Research has cited the combination of Design Thinking and Requirements Engineering, many correlating Design Thinking as a support technique for the elicitation step, however, little is known about the real benefits and challenges that this combination can bring. From the point of view of the development process, there is little empirical evidence of how Design Thinking interactions with Requirements Engineering occur. Given this scenario, this paper aims to understand how Design Thinking practices are applied in each of the Requirements Engineering stages in software projects. To elucidate these interactions, a qualitative and exploratory research was carried out through the application of the case study method in IT organizations in Brazil that work in the development of software projects. The results indicate that Design Thinking has aided Requirements Engineering, both in projects that adopt agile methods and those that adopt the waterfall process, bringing a complementary thought that seeks to build the best software solution design for business problems. It was also possible to conclude that organizations choose to use Design Thinking not based on a specific software family (e.g. mobile or desktop applications), but given the characteristics of the software projects, such as: vague nature of the problem, complex problems and/or need for innovative solutions.

Keywords: Software Engineering, Requirements Engineering, Design Thinking.



## RESUMO

As organizações, impulsionadas pela digitalização dos negócios, têm no software o principal núcleo de geração de valor e o principal canal de comunicação com seus clientes. O software, além de responder às necessidades momentâneas de mercado, abrange uma extensa família de produtos, indo dos aplicativos móveis às plataformas multilaterais. Com isso, a especificação de software precisa representar soluções focadas nos problemas dos consumidores e nas necessidades do mercado. Entretanto, a Engenharia de Requisitos, cuja abordagem é fortemente ligada à tecnologia, torna-se deficitária e pouco efetiva quando o problema não está bem definido ou quando procura-se uma solução inovadora, necessitando assim de uma abordagem complementar. Há pesquisas citando a combinação do Design Thinking com a Engenharia de Requisitos, muitas correlacionando o Design Thinking como uma técnica de apoio à etapa de elicitação, contudo, pouco se conhece sobre os reais benefícios e desafios que esta combinação pode resultar. Sob o ponto de vista do processo de desenvolvimento, existem poucas evidências empíricas de como efetivamente ocorrem as interações do Design Thinking com a Engenharia de Requisitos. Diante deste cenário, este trabalho teve por objetivo compreender como as práticas do Design Thinking são aplicadas em cada uma das etapas da Engenharia de Requisitos em projetos de software. Para elucidar estas interações foi realizada uma pesquisa qualitativa e exploratória, por meio da aplicação do método de estudo de caso em três organizações de tecnologia da informação do Brasil que atuam no desenvolvimento de projetos de software. Os resultados indicam que o Design Thinking tem auxiliado a Engenharia de Requisitos, tanto em projetos que adotam métodos ágeis quanto os que adotam o processo waterfall, trazendo um pensamento complementar que busca construir o melhor desenho de solução de software para os problemas de negócio. Também foi possível concluir que as organizações escolhem aplicar o Design Thinking não por uma família específica de software (ex. aplicações móveis ou desktop) mas sim pelas características dos projetos tais como: natureza vaga do problema, problemas complexos e/ou necessidade de inovação nas soluções.

Palavras-chaves: Engenharia de Software, Engenharia de Requisitos, Design Thinking.

## SUMÁRIO

<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>1</b>
1.1 MOTIVAÇÃO .....	4
1.2 OBJETIVOS DA PESQUISA .....	5
1.3 DELIMITAÇÃO DO ESCOPO .....	5
1.4 MÉTODO DE PESQUISA .....	6
1.5 ESTRUTURA DO DOCUMENTO DA DISSERTAÇÃO .....	6
1.6 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	7
<b>CAPÍTULO 2 – REVISÃO DA LITERATURA .....</b>	<b>8</b>
2.1 REQUISITOS DE SOFTWARE .....	8
<b>2.1.1</b> Requisitos em Sistemas Móveis .....	10
2.2 ENGENHARIA DE REQUISITOS E SUAS TÉCNICAS .....	11
<b>2.2.1</b> Elicitação de Requisitos.....	12
<b>2.2.2</b> Análise e Negociação de Requisitos .....	15
<b>2.2.3</b> Documentação de Requisitos .....	17
<b>2.2.4</b> Validação de Requisitos .....	21
<b>2.2.5</b> Gerenciamento de Requisitos .....	22
<b>2.2.6</b> Engenharia de Requisitos em Métodos Ágeis .....	23
2.3 PROBLEMAS NA ENGENHARIA DE REQUISITOS .....	27
2.4 DESIGN THINKING .....	28
<b>2.4.1</b> Modelos de Design Thinking.....	30
<b>2.4.2</b> Técnicas do Design Thinking.....	33
2.5 DESIGN THINKING E A ENGENHARIA DE SOFTWARE .....	35
2.6 DESIGN THINKING E A ENGENHARIA DE REQUISITOS .....	36
2.7 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	39
<b>CAPÍTULO 3 – MÉTODO DE PESQUISA.....</b>	<b>40</b>
3.1 MÉTODO DE PESQUISA.....	40
3.2 ESTRATÉGIA DE PESQUISA.....	40
<b>3.2.1</b> Questão de Pesquisa Principal .....	42
<b>3.2.2</b> Proposições .....	42
<b>3.2.3</b> Unidades de Análise ou Casos de Estudo .....	43
<b>3.2.4</b> Lógica da Relação entre os Dados e as Proposições .....	43
<b>3.2.5</b> Pontos de Análise .....	44

<b>3.2.6</b>	Relacionamento dos Pontos de Análise com as Proposições .....	49
<b>3.2.7</b>	Critérios para Interpretações dos Dados .....	53
<b>3.2.8</b>	Protocolo de Pesquisa .....	54
<b>3.3</b>	CONSIDERAÇÕES SOBRE O CAPÍTULO .....	55
<b>CAPÍTULO 4 - ESTUDOS DE CASO</b> .....		<b>56</b>
4.1	UNIDADES DE ANÁLISE ESTUDADAS .....	56
4.2	DESCRIÇÃO DA COLETA DE DADOS E TRATAMENTO DOS DADOS .....	57
4.3	ORGANIZAÇÃO A.....	57
<b>4.3.1</b>	Caracterização da Organização A .....	57
<b>4.3.2</b>	Contextualização da Adoção do Design Thinking com a Engenharia de Requisitos na Organização A.....	58
<b>4.3.3</b>	Resultados das Análises das Proposições na Organização A.....	58
4.4	ORGANIZAÇÃO B.....	66
<b>4.4.1</b>	Caracterização da Organização B .....	66
<b>4.4.2</b>	Contextualização da Adoção do Design Thinking com a Engenharia de Requisitos na Organização B.....	67
<b>4.4.3</b>	Resultados das Análises das Proposições na Organização B.....	67
4.5	ORGANIZAÇÃO C.....	74
<b>4.5.1</b>	Caracterização da Organização C .....	74
<b>4.5.2</b>	Contextualização da Adoção do Design Thinking com a Engenharia de Requisitos na Organização C .....	75
<b>4.5.3</b>	Resultados das Análises das Proposições na Organização .....	76
4.6	RESUMO DOS RESULTADOS OBTIDOS NOS ESTUDOS DE CASO .....	82
4.7	CONSIDERAÇÕES SOBRE O CAPÍTULO .....	83
<b>CAPÍTULO 5 - DISCUSSÃO DOS RESULTADOS</b> .....		<b>84</b>
5.1	ANÁLISE DAS PROPOSIÇÕES.....	84
<b>5.1.1</b>	Proposição 1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos.....	85
<b>5.1.2</b>	Proposição 2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis .....	95
<b>5.1.3</b>	Proposição 3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis .....	98
5.2	GENERALIZAÇÃO DOS ESTUDOS DE CASO.....	100

5.3	CONSIDERAÇÕES SOBRE O CAPÍTULO .....	100
<b>CAPÍTULO 6 - CONSIDERAÇÕES FINAIS.....</b>		<b>101</b>
6.1	ANÁLISE DA QUESTÃO DE PESQUISA PRINCIPAL .....	101
6.2	RELEVÂNCIA DO ESTUDO.....	102
6.3	CONTRIBUIÇÕES DA PESQUISA.....	102
6.4	LIMITAÇÕES .....	103
6.5	TRABALHOS FUTUROS .....	103
<b>REFERÊNCIAS .....</b>		<b>104</b>
<b>APÊNDICE A – PROTOCOLO DE PESQUISA – TERMO DE CONFIDENCIALIDADE .....</b>		<b>109</b>
<b>APÊNDICE B – PROTOCOLO DE PESQUISA – APRESENTAÇÃO DA PESQUISA .....</b>		<b>110</b>
<b>APÊNDICE C – PROTOCOLO DE PESQUISA – VISÃO GERAL DA PESQUISA</b>		<b>111</b>

## LISTA DE FIGURAS

Figura 1 - Modelo de alto nível das atividades do processo de engenharia de requisitos, adaptado de (KOTONYA; SOMMERVILLE, 1998). .....	12
Figura 2 - Histórico das técnicas de elicitação de requisitos, adaptado de (JALIL et al., 2019). .....	13
Figura 3 – Diamante duplo: espaço do problema e espaço da solução do Design Thinking, adaptado de (LINDBERG et al., 2011). .....	29
Figura 4 - Modelos de Design Thinking, adaptado de (SOUZA et al., 2017) .....	31
Figura 5 - Três estágios do processo de Design Thinking, adaptado de (BROWN, 2008) ...	32
Figura 6 - Processo de estudos de casos múltiplos, adaptado de (YIN, 2015).....	41

## LISTA DE QUADROS

Quadro 1 - Mapeamento dos processos de Engenharia de Requisitos em abordagens ágeis de desenvolvimento, fonte (O autor, 2020).....	24
Quadro 2 – Análise dos estudos relacionados, fonte (O autor, 2020) .....	38
Quadro 3 - Conceitos de apoio para análise das proposições, fonte (O autor, 2020).....	43
Quadro 4 - Pontos de análise das proposições, fonte (O autor, 2020) .....	46
Quadro 5 - Intersecção entre pontos de análise e proposições, fonte (O autor, 2020).....	51
Quadro 6 – Perfil das unidades de análise e dos entrevistados, fonte (O autor, 2020) .....	56
Quadro 7 – Resultados obtidos no estudo de caso da Organização A, fonte (O autor, 2020) .....	63
Quadro 8 – Nível de influência entre pontos de análise e proposições na Organização A, fonte (O autor, 2020) .....	64
Quadro 9 – Resultados obtidos no estudo de caso da Organização B, fonte (O autor, 2020) .....	71
Quadro 10 – Nível de influência entre pontos de análise e proposições na Organização B, fonte (O autor, 2020) .....	72
Quadro 11 – Resultados obtidos no estudo de caso da Organização C, fonte (O autor, 2020) .....	79
Quadro 12 – Nível de influência entre pontos de análise e proposições na Organização C, fonte (O autor, 2020) .....	80
Quadro 13 – Resultados obtidos nos estudos de caso desta pesquisa, fonte (O autor, 2020) .....	83
Quadro 14 – Síntese da verificação dos pontos de análise nas organizações, fonte (O autor, 2020).....	84
Quadro 15 – Síntese da verificação das proposições nas organizações, fonte (O autor, 2020) .....	85
Quadro 16 – Síntese da proposição P1 nas organizações, fonte (O autor, 2020).....	85
Quadro 17 – Tipos de requisitos levantados pelo Design Thinking, fonte (O autor, 2020) ...	88
Quadro 18 – Tipos de documentos gerados pelo Design Thinking, fonte (O autor, 2020)....	92
Quadro 19 – Abrangência da documentação de requisitos pelo Design Thinking, fonte (O autor, 2020).....	93
Quadro 20 – Síntese da proposição P2 nas organizações, fonte (O autor, 2020) .....	95
Quadro 21 – Síntese da proposição P3 nas organizações, fonte (O autor, 2020).....	98

## LISTA DE ABREVIATURAS E SIGLAS

BABOK	Business Analysis Body of Knowledge Guide
CASE	Computer Aided Software Engineering
DT	Design Thinking
ER	Engenharia de Requisitos
ERA	Engenharia de Requisitos Ágil
ERT	Engenharia de Requisitos Tradicional
IoT	Internet of Things / Internet das Coisas
JAD	Joint Application Design
MVP	Minimum Viable Product
PMI	Project Management Institute
SRS	Software Requirements Specification
SSM	Soft System Methods
TI	Tecnologia da Informação
UX	User Experience

## CAPÍTULO 1 - INTRODUÇÃO

O desenvolvimento de software está se tornando cada vez mais desafiador, motivado pelo aumento de famílias de software que precisam ser produzidas e pelo reposicionamento estratégico que o software passa a representar nas organizações. O software, antes considerado uma ferramenta de suporte, atualmente representa o principal núcleo de geração de valor para as empresas dentro do âmbito de vários segmentos de negócios. Esta tendência está cada vez mais presente em vários tipos de produtos de software, das plataformas multilaterais (Ebay, Uber, Alibaba, Airbnb, Mercado Livre, dentre outros), as quais conectam grupos de pessoas entre si para gerar valor, até os tradicionais aplicativos móveis (*apps*) que permitem agregar valor diretamente aos seus usuários por meio da experiência de uso, muitas vezes processando informações diretamente no celular sem a necessidade de conexão com sistemas de *back-end*.

Essa proximidade dos consumidores finais com o software, aliada com as necessidades momentâneas do mercado, faz com que novas abordagens de especificação de software precisem ser incorporadas à Engenharia de Software, principalmente na Engenharia de Requisitos, permitindo assim encontrar soluções mais focadas nos problemas dos consumidores e nas necessidades do mercado. Entretanto, observa-se que nos processos de desenvolvimento de software, como os processos iterativos e em cascata, para se alcançar um nível adequado de sucesso no desenvolvimento de aplicações faz-se necessário um esforço considerável na fase de Engenharia de Requisitos. Nesta fase há uma maior necessidade de envolvimento dos usuários, para garantir que todas as funcionalidades e restrições estejam bem detalhadas e documentadas, suportando a condução das fases subsequentes do desenvolvimento (KOTONYA; SOMMERVILLE, 1998).

Para que a Engenharia de Requisitos seja eficiente é necessário que, além da escolha de técnicas adequadas para o tratamento da complexidade de se trabalhar com diversos usuários, seja também disponibilizado tempo suficiente para a execução do processo, o que nem sempre está presente no orçamento e no cronograma dos projetos, sendo imprescindível para a construção de sistemas de informação grandes



e complexos. No entanto, as técnicas tradicionais da Engenharia de Requisitos precisam também lidar com o aumento significativo de aplicativos móveis (*apps*) os quais são pequenos, requerem rápido desenvolvimento, precisam se ajustar fortemente com as necessidades dos usuários e mudam constantemente (VETTERLI et al., 2013).

O grande desafio que as comunidades de desenvolvimento de software têm tentado solucionar é o de como a partir do espaço do problema (requisitos do usuário) alcançar o espaço da solução (requisitos de sistema), guiado por um método (VETTERLI et al., 2013), sendo a Engenharia de Requisitos um exemplo de processo padronizado para resolver este problema (VETTERLI et al., 2013). Por outro lado, quando o espaço do problema não é bem definido, ou procura-se uma solução inovadora, a Engenharia de Requisitos torna-se deficitária e pouco efetiva, pois sua abordagem de convergência (HEHN; UEBERNICKEL, 2018) nem sempre alcança os melhores resultados nos produtos desenvolvidos.

O surgimento dos métodos ágeis no desenvolvimento de software mudaram a perspectiva do processo tradicional de Engenharia de Requisitos, por meio de ciclos mais curtos para as entregas e trazendo um maior envolvimento dos usuários durante toda a fase de concepção do software, por meio de uma abordagem fortemente ligada à prototipação, baixa documentação de requisitos e grande rastreabilidade de código. Contudo, os métodos ágeis têm uma tendência de evitar a exploração de diversas soluções para um mesmo problema de software, mantendo sempre uma visão geral do que se pretende fazer em seguida (LINDBERG et al., 2011), o que limita a implementação de soluções inovadoras.

A combinação das melhores práticas da Engenharia de Requisitos com as melhores práticas dos métodos ágeis daria luz à emergente necessidade de desenvolvimento rápido de novas aplicações, crucial para os negócios, e cada vez mais presente no dia a dia das empresas e dos usuários (VETTERLI et al., 2013). Nesta direção, de acordo com (LINDBERG et al., 2011), se a Engenharia de Requisitos utilizar os conceitos e práticas do Design Thinking para a criação de software, por meio de uma abordagem centrada no usuário e não mais na tecnologia, permitirá não somente uma maior velocidade nas soluções de problemas complexos dos softwares atuais como também ampliará e enriquecerá as técnicas tradicionais de elicitação de requisitos (CARROLL; RICHARDSON, 2016). Isso acontece porque o Design Thinking é reconhecido como uma metodologia estruturada para solução de

problemas, a qual pode ser aplicada em times multidisciplinares por meio da exploração das necessidades humanas e da rápida prototipação em ciclos iterativos durante os primeiros estágios de concepção de um produto, serviço ou sistema (HEHN; UEBERNICKEL, 2018).

A aplicação da metodologia do Design Thinking no desenvolvimento de software surge como um pensamento complementar (LINDBERG et al., 2011), que permite resolver problemas complexos a partir de uma abordagem mais iterativa e centrada no usuário. O Design Thinking se apresenta como uma metodologia para elicitare as necessidades dos consumidores, ao invés de apenas produzir uma listagem de requisitos, gerando como resultado uma série de protótipos que eventualmente convergem para uma solução inovadora (VETTERLI et al., 2013). A abordagem de divergência do Design Thinking possibilita encontrar muitas vezes soluções inovadoras para um problema, diferentemente da abordagem tradicional de convergência das práticas da Engenharia de Requisitos (HEHN; UEBERNICKEL, 2018), que além de estarem focadas na tecnologia têm o enfoque voltado à resolução e detalhamento de requisitos para problemas já definidos. O Design Thinking utiliza de forma sistemática esta abordagem para explorar as definições dos problemas e sintetizar soluções dentro de um processo que abrange inspiração, ideação e implementação (BROWN, 2008).

Há uma tendência de relacionamento do Design Thinking com as atividades iniciais de elicitação da Engenharia de Requisitos (VETTERLI et al., 2013; SOUZA; SILVA, 2015; SOUZA et al., 2018; HEHN et al., 2020). O desafio em se descobrir e satisfazer as necessidades, muitas vezes confusas e voláteis, das partes interessadas nos projetos de software, fazem com que a abordagem centrada no usuário e com um método de geração rápida de protótipos do Design Thinking seja algo promissor para responder à estes desafios (HEHN et al., 2020). Estudos acadêmicos existentes incentivam esses esforços e reconhecem o Design Thinking não somente como um suporte para as práticas de Engenharia de Requisitos, mas como uma forma moderna de Engenharia de Requisitos (VETTERLI et al., 2013; HEHN; UEBERNICKEL, 2018).

A partir deste cenário, percebe-se um inter-relacionamento entre as práticas do Design Thinking com o processo de Engenharia de Requisitos e este será o objeto desta pesquisa, a qual promoverá uma compreensão de como estas duas áreas estão correlacionadas e qual é a contribuição que este relacionamento gera para o desenvolvimento de software.

## 1.1 Motivação

Estudos recentes, como o de (HEHN; UEBERNICKEL, 2018), demonstram que existe potencial na combinação do Design Thinking (uma abordagem mais centrada no usuário) com a Engenharia de Requisitos (uma abordagem mais formal e direcionada pela tecnologia) podendo apresentar ganhos efetivos para o desenvolvimento de software, como por exemplo, o incremento na qualidade do software, velocidade no processo de desenvolvimento e melhoria na aceitação do usuário.

Diversos estudos como os de (VETTERLI et al., 2013; ARAÚJO et al., 2015; CARROL; RICHARDSON, 2016; SOUZA et al., 2017) demonstram que a Engenharia de Software já faz uso das práticas do Design Thinking para auxiliar a Engenharia de Requisitos. Isto é aplicado principalmente nas problemáticas inerentes dos projetos que se propõem desenvolver soluções inovadoras com alto grau de aderência às necessidades dos usuários, buscando uma abordagem interativa com rápido tempo de resposta. O estudo de (HEHN et al., 2020) demonstra a integração da Engenharia de Requisitos com o Design Thinking, onde o Design Thinking é utilizado para primeiramente realizar um melhor entendimento do problema, para que posteriormente a Engenharia de Requisitos promova a documentação dos requisitos da solução.

Entretanto, apesar de serem demonstradas as aplicações do Design Thinking no desenvolvimento de software em diversas áreas, (HEHN; UEBERNICKEL, 2018) afirmam que pouco se conhece sobre os reais benefícios e desafios que esta combinação pode resultar. Além disso, existem poucas evidências empíricas que demonstrem o nível de profundidade e de correlação da aplicação da metodologia do Design Thinking dentro do processo de Engenharia de Requisitos.

Neste contexto, este trabalho se propõe a investigar **como as práticas do Design Thinking são aplicadas na Engenharia de Requisitos em projetos de software**, analisando sob a ótica dos atores envolvidos no processo como cada uma das etapas da fase de Engenharia de Requisitos se beneficiam e/ou se correlacionam, direta ou indiretamente, com as práticas do Design Thinking. A pesquisa também apresentará uma análise da interação entre estas duas áreas, sendo possível melhorar a compreensão da sua combinação, bem como analisar possíveis impactos e implicações em sua adoção no desenvolvimento de projetos de software, auxiliando

gestores e equipes de desenvolvimento em outras organizações na adoção do Design Thinking na Engenharia de Requisitos.

O resultado deste trabalho é relevante para a pesquisa na Engenharia de Software pois trará elucidação da interação entre o processo de Engenharia de Requisitos e a metodologia de Design Thinking, considerando sua aplicação em projetos de software no Brasil, cuja utilização ainda é pouco conhecida.

## 1.2 Objetivos da pesquisa

Diante do cenário apresentado, este trabalho tem como objetivo geral: **Compreender como as práticas do Design Thinking são aplicadas na Engenharia de Requisitos em projetos de software.**

Para atingir o objetivo geral, foram delineados os seguintes objetivos específicos:

- I. Identificar as práticas do Design Thinking que estão sendo utilizadas na Engenharia de Requisitos; e
- II. Analisar as práticas identificadas sob a ótica da Engenharia de Requisitos.

Desta forma será possível responder à questão principal desta pesquisa: **Como o Design Thinking auxilia a Engenharia de Requisitos em projetos de software?**

## 1.3 Delimitação do escopo

Esta pesquisa destina-se a compreender e aprofundar questões sobre como as organizações utilizam o Design Thinking como suporte para o desenvolvimento de projetos de software. Não faz parte deste estudo aprofundar as questões de efetividade da aplicação do método na resolução de problemas e ou investigar se os métodos e vertentes de Design Thinking estão corretamente sendo aplicados nestas organizações.

As empresas, objeto desse estudo, são empresas de TI que utilizam o Design Thinking para a criação de soluções de software. Esta pesquisa aborda somente os indivíduos que participam do processo de desenvolvimento de software, como os profissionais de Design Thinking, User Experience, Arquitetos, Analistas de Sistemas e Gerentes de Desenvolvimento.

## 1.4 Método de pesquisa

Para organizar o trabalho de pesquisa a ser realizado, foi definido um planejamento inicial composto por três fases:

- **Fase 1 – Preparação e estruturação da pesquisa:** esta fase corresponde à delimitação da área de estudo, coleta e análise das referências bibliográficas, delimitação do tema, estabelecimento dos objetivos e questões de pesquisa, elaboração de um quadro referencial teórico, seleção do método de pesquisa e definição das etapas de pesquisa.
- **Fase 2 – Execução da pesquisa:** fase da investigação com aplicação do método de estudo de caso para múltiplos casos segundo a abordagem proposta por (YIN, 2015), com coleta de dados em campo por meio de entrevistas semiestruturadas em empresas que utilizam o Design Thinking no desenvolvimento de software.
- **Fase 3 - Análise dos resultados:** análise dos dados de forma individual e agrupada, permitindo a geração de generalizações e das conclusões do trabalho.

## 1.5 Estrutura do documento da dissertação

Este documento está organizado com a seguinte estrutura:

- Capítulo 1 apresentado aqui tem por objetivo demonstrar ao leitor um panorama geral sobre os objetivos e motivações para esta pesquisa.
- Capítulo 2 apresenta um aprofundamento teórico dos conceitos da Engenharia de Requisitos e do Design Thinking.
- Capítulo 3 apresenta o método de pesquisa que embasará a condução da pesquisa, com o detalhamento de cada uma das etapas.
- Capítulo 4 descreve os estudos de caso realizados em três organizações, bem como as respectivas análises.
- Capítulo 5 apresenta uma discussão dos resultados identificados nos estudos de casos conduzidos por esta pesquisa, correlacionando-os ao referencial teórico, às proposições e aos respectivos conceitos de apoio.

- Capítulo 6 apresenta a conclusão desta pesquisa, a relevância deste estudo, sua contribuição e limitações e trabalhos futuros.

### **1.6 Considerações sobre o capítulo**

Este capítulo apresentou os principais aspectos que motivaram este estudo para uma melhor compreensão de como o Design Thinking está relacionado com a Engenharia de Requisitos em projetos de software.

## CAPÍTULO 2 – REVISÃO DA LITERATURA

Este capítulo apresenta o referencial teórico das áreas de Engenharia de Requisitos e Design Thinking, com suas principais técnicas. Também aborda as pesquisas relacionadas a este trabalho.

### 2.1 Requisitos de Software

Diversas interpretações e classificações de requisitos estão presentes na literatura. (KOTONYA; SOMMERVILLE, 1998) descrevem que requisitos são definidos nos primeiros estágios do desenvolvimento de um sistema, como uma especificação do que deverá ser implementado. Para (PAETSCH et al., 2003) requisitos descrevem o que deve ser feito por uma aplicação, mas não descrevem como serão implementados no sistema. Os requisitos conceituam as especificações como descrições de como o sistema deve se comportar, qual domínio de informação da aplicação, quais restrições a aplicação deve ter em sua operação ou podem ser especificações de uma propriedade ou atributo do sistema. Algumas vezes podem ser considerados como restrições no processo de desenvolvimento (KOTONYA; SOMMERVILLE, 1998).

Definir requisitos é o processo de registrar, por meio de alguma forma documental, a representação de uma determinada condição ou capacidade exigida por um usuário em um sistema para que este possa resolver um problema. Pode ser considerado também como a representação de uma determinada condição que permita ao usuário de uma aplicação atingir um determinado objetivo. Para (KOTONYA; SOMMERVILLE, 1998) requisitos são uma declaração de um serviço de sistema ou uma declaração de uma restrição de sistema.

A classificação da natureza dos requisitos, a partir da perspectiva do usuário, pode ser: aparente ou oculto, conhecido ou desconhecido, esperado ou não esperado (CARROLL; RICHARDSON, 2016). A classificação dos requisitos, a partir da perspectiva do desenvolvimento, pode ser (SOMMERVILLE, 2019):

- Requisitos do usuário: são declarações em linguagem natural, eventualmente com a complementação de diagramas, dos serviços que o

sistema deve prover e suas restrições operacionais. Representam as necessidades e restrições desejadas pelo usuário; e

- Requisitos do sistema: documentações estruturadas e detalhadas das funções, serviços e restrições operacionais do sistema. Estas documentações estabelecem um contrato que define o que precisa ser implementado para se atingir os objetivos do usuário.

De acordo com (SOMMERVILLE, 2019) os requisitos do sistema podem ser subclassificados em três categorias:

- Requisitos funcionais: declarações dos serviços que o sistema deve prover, como o sistema deve reagir à determinadas entradas e como o sistema deve se comportar em situações particulares;
- Requisitos de domínio: requisitos que surgem do domínio da aplicação e que refletem este domínio; e
- Requisitos não-funcionais: descrevem restrições nos serviços ou funções oferecidas pelo sistema como restrições de tempo de resposta, restrições de desenvolvimento, processos, padrões, dentre outros.

Os requisitos não-funcionais também podem passar por uma subclassificação delimitada por grupos de afinidade. Diferentes formas de classificar requisitos não-funcionais já foram propostas (KOTONYA; SOMMERVILLE, 1998). A norma IEEE Std 830-1993 (IEEE, 1993) propõe uma classificação de requisitos não-funcionais composta por 13 categorias: Performance, Interface, Operacional, Recursos, Verificação, Aceitação, Documentação, Segurança, Portabilidade, Qualidade, Confiabilidade, Manutenibilidade e Segurança. Já (ROMAN, 1985) propõe uma taxonomia para requisitos não-funcionais composta por 6 classes: Interface, Desempenho, Operação, Ciclo de vida, Econômico e Político.

Por outro lado, (SOMMERVILLE, 2019) propõe uma classificação de requisitos não-funcionais agrupada em 3 grandes grupos: Processo/Organizacional, Produtos e Externos.

A mais abrangente classificação para requisitos não-funcionais foi apresentada por (CHUNG et al., 2000) a qual inclui mais de 150 classes diferentes de requisitos não-funcionais (BAKER et al., 2019).

O levantamento e documentação de todos os requisitos de um projeto de software, bem como suas respectivas classificações, ocorrerão por meio do processo



de Engenharia de Requisitos o qual aplica diversas técnicas para auxiliar a equipe de desenvolvimento de software na descoberta, documentação e validação de requisitos.

Os aplicativos móveis (*apps*) cada vez mais presentes no cotidiano das empresas e das pessoas derivaram particularidades específicas tanto na definição como na classificação dos requisitos para esta família de software.

### **2.1.1 Requisitos em Sistemas Móveis**

Os sistemas móveis são distintos dos sistemas tradicionais. Enquanto os sistemas tradicionais rodam em estações de trabalho e/ou servidores, com ambientes mais controlados, os sistemas móveis rodam em dispositivos que nem sempre estão conectados a um servidor. Tais sistemas têm que se adaptar à ambientes heterogêneos, com recursos limitados, provendo valor para o usuário. A usabilidade é mais difícil de ser garantida em sistemas móveis devido à limitação física dos dispositivos e a experiência do usuário é importante para promover o posicionamento do produto no mercado, uma vez que estas aplicações podem rodar em milhões de dispositivos. Todas estas características são consideradas essenciais na análise de requisitos de projetos de software móvel.

Como no desenvolvimento de software tradicional, o desenvolvimento de software móvel deve cumprir com objetivos claros e aplicar práticas comuns para garantir o seu sucesso. Entretanto, novas restrições, incomuns nos sistemas desktops tradicionais, precisam ser endereçadas neste tipo de aplicação, como por exemplo: problemas de comunicação wireless, questões de mobilidade, variedade de plataformas e tecnologias, as limitações de processamento dos dispositivos, potencial iterações com outros aplicativos, manipulação de sensores, ser uma aplicação nativa ou híbrida, funcionar em determinadas famílias de plataformas de hardware e software, características específicas de segurança, interface, complexidade de testes, consumo de energia e requisitos de *time-to-market* (WASSERMAN, 2010; CORRAL et al., 2013).

Nestes ambientes estas restrições nada mais são do que novos contextos de análise para os requisitos não-funcionais dentro das 3 categorias propostas por (SOMMERVILLE, 2019): processo/organizacional, produto e externo.

Para (DEHLINGER; DIXON, 2011) os requisitos não-funcionais são críticos nos sistemas móveis e pontuam que algumas aplicações móveis podem necessitar se auto adaptar dinamicamente, fornecendo funcionalidades reduzidas. Com isto, novas

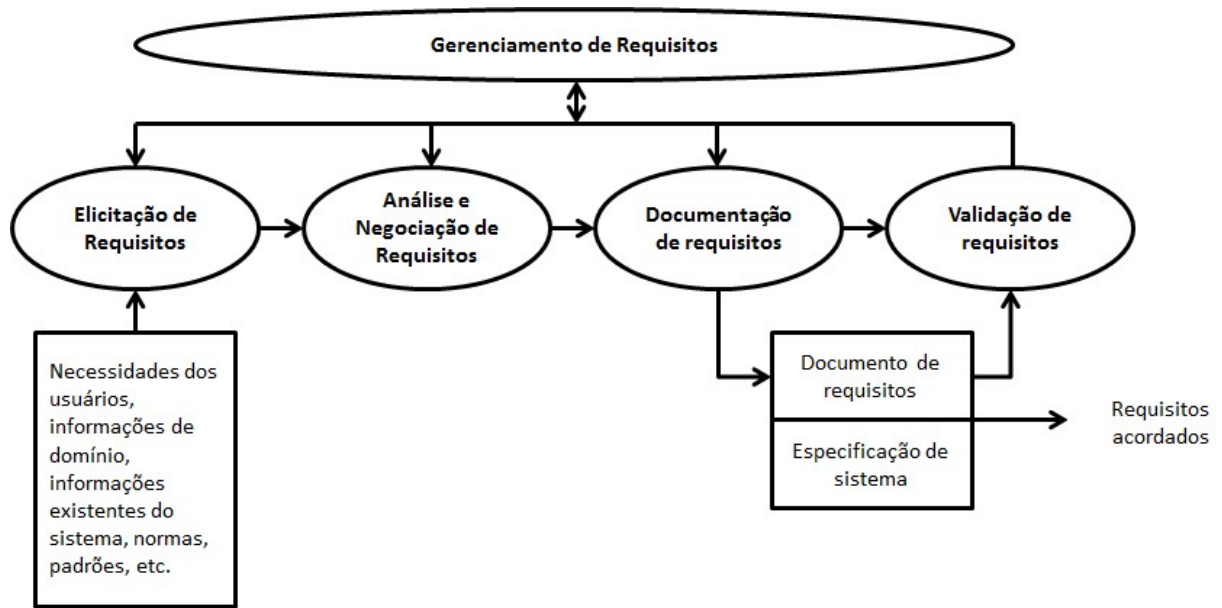
abordagens de especificação de requisitos para sistemas móveis auto adaptáveis precisam ser incorporadas à Engenharia de Requisitos, como por exemplo a utilização da linguagem RELAX (WHITTLE et al., 2010).

## **2.2 Engenharia de Requisitos e suas técnicas**

A Engenharia de Requisitos, uma das fases mais importantes da Engenharia de Software, tem sido utilizada ao longo dos anos para a identificação e documentação de escopo de diversas plataformas de software, dentro dos mais variados contextos de utilização, tendo como foco a delimitação dos objetivos, das necessidades, das funcionalidades e das restrições que uma aplicação de software necessita apresentar dentro de um domínio de utilização. Para (PAETSCH et al., 2003) o principal objetivo da Engenharia de Requisitos é identificar, modelar, comunicar e documentar os requisitos para um sistema, bem como determinar os contextos em que a aplicação será utilizada.

O termo Engenharia de Requisitos foi criado para envolver todas as atividades relacionadas à descoberta, documentação e manutenção de um conjunto de requisitos para um sistema computacional. O uso da terminologia “engenharia” implica que técnicas sistemáticas e repetíveis deverão ser utilizadas para assegurar que todos os requisitos de sistema são completos, consistentes e relevantes (KOTONYA; SOMMERVILLE, 1998). A Engenharia de Requisitos é conhecida como a primeira fase do processo de desenvolvimento de software e é considerada como a principal fase deste processo (RAMINGWONG, 2012).

Para (KOTONYA; SOMMERVILLE, 1998) o processo de Engenharia de Requisitos é um processo de design, pois envolve criatividade, interações entre um extenso e heterogêneo grupo de pessoas, julgamentos, conhecimento prévio de engenharia e experiência. Este processo está estruturado como um conjunto de atividades com o objetivo de criar, validar e manter a documentação dos requisitos de sistema e é composto por quatro atividades principais: elicitação/descoberta de requisitos, análise e negociação de requisitos, documentação de requisitos e validação de requisitos (KOTONYA; SOMMERVILLE, 1998; PAETSCH et al., 2003; RAMINGWONG, 2012; VETTERLI et al., 2013; CARROLL; RICHARDSON, 2016) e tendo o suporte de uma atividade denominada gerenciamento de requisitos (KOTONYA; SOMMERVILLE, 1998). O desenho deste processo está demonstrado pela Figura 1:

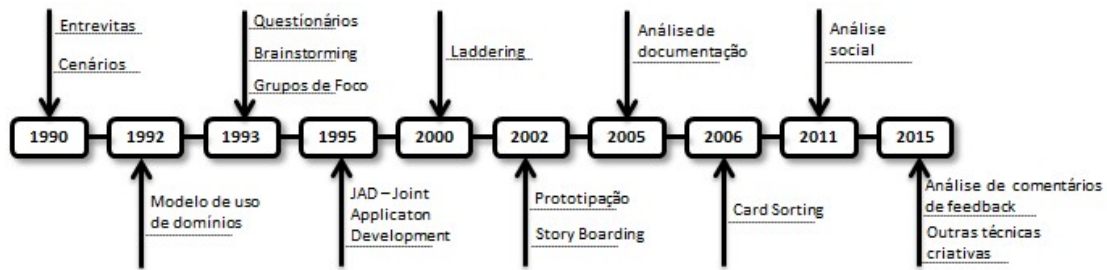


**Figura 1 - Modelo de alto nível das atividades do processo de engenharia de requisitos, adaptado de (KOTONYA; SOMMERVILLE, 1998).**

### 2.2.1 Elicitação de Requisitos

Na atividade de Elicitação de Requisitos os requisitos de sistema são descobertos por meio de entrevistas com as partes interessadas, documentos de sistema, conhecimento do domínio e estudos de mercado (KOTONYA; SOMMERVILLE, 1998). Para (RAMINGWONG, 2012) a elicitação tem como principal objetivo determinar o propósito do sistema que será desenvolvido. A elicitação de requisitos é considerada a atividade mais complexa da Engenharia de Requisitos (SOUZA et al., 2018) e tem por objetivo descobrir as necessidades das partes interessadas que direta ou indiretamente influenciam na determinação dos requisitos do sistema. Já (KOTONYA; SOMMERVILLE, 1998) descreve que a atividade de elicitação de requisitos e a atividade de análise e negociação de requisitos estão diretamente ligadas, pois conforme os requisitos são identificados alguma análise é inevitavelmente realizada bem como o reconhecimento de problemas que precisam ser tratados para o correto entendimento do requisito.

Os requisitos vão sendo identificados por meio do resultado da aplicação de diversas técnicas de elicitação. Várias técnicas de elicitação de requisitos foram propostas na literatura (JALIL et al., 2019). O estudo de (JALIL et al., 2019) demonstrou a evolução das técnicas de elicitação em uma linha do tempo conforme apresentado pela Figura 2:



**Figura 2 - Histórico das técnicas de elicitação de requisitos, adaptado de (JALIL et al., 2019).**

As técnicas de elicitação são agrupadas em categorias, conforme o seu contexto de aplicação. Para (JALIL et al., 2019) existem cinco categorias para classificação das técnicas de elicitação conforme sua natureza de comunicação:

1. Técnicas Tradicionais: composta pelas técnicas de Entrevistas, Questionários, Coleta de dados de sistemas existentes e *Survey*;
2. Técnicas Contextuais: composta pelas técnicas de Observação e Etnografia/Análise Social;
3. Técnicas Colaborativas/Grupos: composta pelas técnicas de *Laddering*, Grupos Focais, *Brainstorming*, *JAD*, Prototipação, *Workshop*, *Story Boarding*, Modelos e Cenários do Usuário;
4. Técnicas Cognitivas: Análise de Documentos, *Card Sorting*, *Protocol Analysis*, *Laddering* e *Repository grid*;
5. Técnicas Inovadoras: composta pela técnica *Throwaway Paper Prototype* (KUMAR et al., 2019).

Já (NUSEIBEH; EASTERBROOK, 2000) classificam as técnicas de elicitação de requisitos em seis categorias conforme demonstrado no estudo de (RAMINGWONG, 2012):

1. Técnicas tradicionais: são amplamente estabelecidas e bem conhecidas, como Análise Documental, Questionários, Entrevistas e Reuniões;
2. Técnicas de elicitação em grupos: incentivam as partes envolvidas a comunicar de forma mais tranquila os requisitos. As principais técnicas deste grupo são Sessões de Grupos Focais e *Brainstorming*;

3. Prototipação: é uma das técnicas mais utilizada para elicitar requisitos ocultos e para analisar *feedbacks*. Normalmente é utilizada com uma combinação de outras técnicas;
4. Técnicas contextuais: envolve estudar os usuários no contexto e o ambiente de implementação do sistema, por meio de Observações e Análise de Diálogos;
5. Técnicas guiadas por modelos: incluem um modelo e um tipo de informação específica para guiar o processo de elicitação; e
6. Técnicas cognitivas: utilizadas para obter conhecimento por meio de várias técnicas.

Conforme (PAETSCH et al., 2003; SOUZA et al., 2018; JALIL et al., 2019) a técnica mais comumente utilizada para elicitação de requisitos é a Entrevista, a qual é um método para descobrir fatos e opiniões de usuários potenciais e das partes interessadas no desenvolvimento do sistema, podendo ser entrevistas fechadas (com um conjunto pré-definido de perguntas) ou abertas (cujo conjunto de perguntas não é pré-determinado).

A técnica de Cenários descrita por (KOTONYA; SOMMERVILLE, 1998; PAETSCH et al., 2003) descreve através de uma história como o sistema será utilizado.

A técnica *Soft system methods* (SSM) descrita por (KOTONYA; SOMMERVILLE, 1998) tem por principal objetivo produzir um modelo menos formal do sistema dentro de um contexto sócio-técnico. A essência do SSM é reconhecer que o sistema está fortemente inserido em um contexto humano e organizacional, o qual provê meios de compreender requisitos abstratos do sistema através da análise do contexto organizacional, do problema a ser resolvido e de sistemas que já existam neste ambiente. Este método introduz a noção de pontos de vista, aonde diferentes pontos de vista apresentam diferentes percepções do problema e dos requisitos de uma solução.

A técnica de Observação e Análise Social descrita por (KOTONYA; SOMMERVILLE, 1998; PAETSCH et al., 2003; JALIL et al., 2019) tem por principal foco utilizar uma abordagem passiva no contexto de uso da aplicação, observando como as atividades são feitas e compreendendo questões culturais e comportamentais, para a identificação de requisitos.

A técnica de Grupo Focal (PAETSCH et al., 2003) é aplicada de maneira informal em pequenos grupos com diferentes conhecimentos e habilidades para que façam uma discussão sobre problemas e percepções sobre os recursos apresentados no protótipo de sistema.

A técnica de Prototipação descrita por (KOTONYA; SOMMERVILLE, 1998; PAETSCH et al., 2003; JALIL et al., 2019) tem por objetivo gerar visualizações preliminares do sistema durante a etapa de especificação de requisitos. Esta técnica permite elicitar e validar os requisitos do sistema.

A técnica de *Brainstorming* (SOUZA et al., 2018; PAETSCH et al., 2003) popularizada em 1953 por Alex Osborn, propõe que um grupo tente encontrar uma solução para o problema específico através da acumulação e geração espontânea de ideias de seus membros.

A técnica de Questionários (SOUZA et al., 2018; JALIL et al., 2019) é uma técnica utilizada para a extração de requisitos conflituosos e inconsistentes dos usuários.

A técnica de sessões de *Joint Application Design* (JAD) (PAETSCH et al., 2003; SOUZA et al., 2018; JALIL et al., 2019) é implementada pela execução de *workshops* coletivos de alto nível, com duração de 4 a 5 dias, cujo principal resultado é uma lista ordenada por sensibilidade sobre as necessidades dos clientes.

A técnica de *Card Sorting* (JALIL et al., 2019) estabelece um método para reunir requisitos onde as partes interessadas são solicitadas a organizar cartões de acordo com nomes de entidades de domínio usando cartões de arquivo ou um número de pacotes de produtos.

A técnica de *Laddering* (JALIL et al., 2019) é realizada por meio de uma entrevista estruturada na qual um conjunto padronizado de perguntas é aplicado para elicitar os objetivos, qualidades e características das partes envolvidas.

Após os requisitos serem elicitados eles passam para a atividade de Análise e Negociação da Engenharia de Requisitos.

### **2.2.2 Análise e Negociação de Requisitos**

Análise e Negociação de Requisitos são atividades com o objetivo de descobrir problemas nos requisitos de sistema e negociar as mudanças necessárias por meio de acordos de forma a satisfazer todas as partes interessadas (KOTONYA;

SOMMERVILLE, 1998). Alguns aspectos são verificados nesta atividade para cada requisito (PAETSCH et al., 2003):

- Necessidade: a real necessidade do requisito;
- Consistência: os requisitos não podem ser contraditórios;
- Completude: nenhum serviço, informação ou restrição foi omitida na especificação do requisito; e
- Viabilidade: requisitos precisam ser viáveis de serem implementados no desenvolvimento do sistema dentro do contexto de orçamento e cronograma.

Conflitos em requisitos são solucionados através de priorização e negociação com as partes envolvidas. Os modelos de negociação ajudam os praticantes a definir os objetivos e estabelecer os mecanismos necessários para assegurar que estes objetivos serão satisfeitos (RAMINGWONG, 2012).

As principais técnicas utilizadas para análise de requisitos são sessões JAD (descrita no Capítulo 2.2.1), Priorização de Requisitos e Modelagem (PAETSCH et al., 2003).

A técnica de Priorização de Requisitos (PAETSCH et al., 2003) é realizada com o envolvimento dos clientes e dos desenvolvedores, os quais precisam fornecer informações sobre cada requisito. Os clientes proveem informações de quais requisitos possuem o maior benefício para os usuários com as maiores prioridades. Os desenvolvedores apontam os riscos técnicos, custo e dificuldades. Com base nestas informações é feito um cruzamento para buscar priorizar os requisitos que possuem o maior benefício com o menor impacto/risco para o desenvolvimento. Existem vários modelos de priorização como, por exemplo, o modelo de emparelhamento ou o *Analytic Hierarchy Process* (KARLsoon; RYAN, 1997).

A técnica de Modelagem (PAETSCH et al., 2003) serve como uma importante conexão entre a análise e o projeto do sistema. Nesta técnica os requisitos são representados por modelos. Os modelos mais populares de representação são os modelos de fluxo de dados, modelos de dados semânticos e os modelos orientados à objeto.

Uma vez finalizada a análise dos requisitos estes passam para a próxima atividade que é a de Documentação de Requisitos.

### 2.2.3 Documentação de Requisitos

Na atividade Documentação de Requisitos os requisitos são cuidadosamente documentados, podendo ser de forma textual, gráfica, simbólica ou uma combinação destas formas (RAMINGWONG, 2012).

O propósito da documentação de requisitos é comunicar requisitos entre as partes interessadas e entre as equipes de desenvolvimento de sistema. Os documentos de requisitos servirão como linha de base para a avaliação dos produtos e processos subsequentes (projeto, teste, atividades de validação e verificação) e para o processo de controle de mudança (PAETSCH et al., 2003).

Nesta fase podem ser produzidas documentações mais detalhadas como, por exemplo, os modelos de sistema (KOTONYA; SOMMERVILLE, 1998). No estudo conduzido por (JARZEBOWICZ; POLOCKA, 2017) foram identificadas 30 técnicas de documentação de requisitos extraídas do *Business Analysis Body of Knowledge Guide* (BABOK, 2011) e do *Business Analysis for Practicioners – A Practice Guide* (PMI GUIDE, 2015) do *Project Management Institute*:

- **Análise de Regras de Negócio (catálogo de regras de negócio):** utilizado para identificar, expressar, validar, refinar e organizar as regras que conduzem o dia a dia do negócio e guiam a tomada de decisão dos negócios (JARZEBOWICZ; POLOCKA, 2017).
- **Dicionário de Dados / Glossário:** é utilizado para padronizar as definições de cada elemento de dados e habilitar a interpretação destes elementos entre as diversas partes interessadas (JARZEBOWICZ; POLOCKA, 2017).
- **Diagrama de Fluxo de Dados:** demonstra os movimentos de transformação dos dados entre as entidades externas e entre os processos (JARZEBOWICZ; POLOCKA, 2017).
- **Modelagem de Dados (diagrama de entidade-relacionamento):** descrevem entidades, classes e objetos de dados, relevantes em um domínio, e todos os atributos que os descrevem além dos relacionamentos entre estes componentes (JARZEBOWICZ; POLOCKA, 2017).



- Modelagem de Decisão (tabelas de decisão, árvores de decisão): demonstram como os dados e conhecimentos são combinados para tomar uma determinada decisão (JARZEBOWICZ; POLOCKA, 2017).
- Decomposição Funcional (modelos de decomposição): ajudam o gerenciamento da complexidade e reduzem incertezas por meio da decomposição de sistemas e conceitos em partes constituintes menores, permitindo que estas partes possam ser analisadas individualmente (JARZEBOWICZ; POLOCKA, 2017).
- Análise de *Interface* (tabelas de *interface* do sistema, fluxo de *interface* do usuário): é utilizado para identificar aonde, o que, porque, quando, como e qual informação é compartilhada entre os componentes de solução ou através das fronteiras da solução (JARZEBOWICZ; POLOCKA, 2017).
- Modelagem Organizacional (organograma): é uma representação visual que define como uma organização, ou uma unidade organizacional, é estruturada (JARZEBOWICZ; POLOCKA, 2017).
- Modelagem de Processo (fluxo de processos): descrevem fluxos sequenciais de trabalhos ou atividades ou processos de sistema (JARZEBOWICZ; POLOCKA, 2017).
- Prototipação: é a representação de um sistema utilizado para validar requisitos elicitados e identificar omissões ou requisitos identificados incorretamente (JARZEBOWICZ; POLOCKA, 2017).
- Análise de Causa Raiz (diagrama de espinha de peixe): é utilizado para identificar e analisar causas subjacentes de um problema ou de uma oportunidade (JARZEBOWICZ; POLOCKA, 2017).
- Modelagem de Escopo (diagrama de contexto): define a natureza de uma ou mais fronteiras e coloca elementos dentro e fora destas fronteiras. Modelagem de escopo é tipicamente representada como uma combinação de diagramas, métricas e explicações textuais (JARZEBOWICZ; POLOCKA, 2017).
- Modelagem de Estado (tabela de estado, diagrama de estado): é utilizada para descrever e analisar diferentes possibilidades de estado de um determinado objeto, permitindo transições de um estado para

outro, bem como atividades internas dentro de um mesmo estado (JARZEBOWICZ; POLOCKA, 2017).

- Casos de Uso e Cenários: descrevem como uma pessoa ou sistema, denominado atores, interagem com a solução que está sendo modelada para atingir um determinado objetivo. Cenários são descritos utilizando textos estruturados através de uma série de passos executados pelos atores ou pela solução. Um caso de uso descreve vários cenários. Um diagrama de casos pode ser utilizado para visualizar relacionamentos entre casos de uso ou entre casos de uso e atores (JARZEBOWICZ; POLOCKA, 2017).
- Estórias do Usuário: representam uma pequena declaração consistente de uma funcionalidade necessária para entregar valor para uma determinada parte interessada (JARZEBOWICZ; POLOCKA, 2017).
- Critérios de Avaliação e Aceitação: definir os requisitos que devem ser atendidos para que a solução seja considerada aceitável pelas partes interessadas. Esta técnica determina quais requisitos devem ser atendidos para que a solução seja considerada aceitável pelas partes interessadas (BABOK, 2011).
- Análise de Capacidade do Negócio: é uma abordagem para descobrir a estrutura do negócio, identificando as unidades de negócio e quantificando a capacidade do negócio para atingir um resultado ou objetivo específico. A análise serve como base para a condução de projetos de melhorias (KLINKMKÜLLER et al., 2010).
- *Business Model Canvas*: pode ser descrito através de nove blocos básicos que demonstram a lógica de como os negócios pretendem gerar receita. Os nove blocos compõem as quatro áreas principais de negócio: clientes, ofertas, infraestrutura e viabilidade financeira (BABOK, 2011).
- Modelagem Conceitual: trata da descrição da semântica de aplicativos de software, no mais alto nível de abstração e não leva em conta o banco de dados em si, mas a forma como as estruturas são organizadas para se armazenar os dados (BABOK, 2011).

- **Análise de Requisitos Não-Funcionais:** descreve as qualidades requeridas para um sistema, como usabilidade e características de desempenho. Eles complementam a documentação de requisitos funcionais que descrevem o comportamento do sistema (BABOK, 2011).
- **Matriz de Papéis e Permissões / Modelagem Organizacional:** a modelagem organizacional é utilizada para descrever os papéis, as responsabilidades e hierarquia de reportes existentes em uma organização alinhando essas estruturas com as metas da organização. Definem como uma organização ou unidade organizacional está estruturada (BABOK, 2011).
- **Diagrama de Sequência:** são utilizados para modelar a lógica de cenários de uso através da apresentação da informação passada entre objetos no sistema, a partir da execução do cenário. Um diagrama de sequência apresenta como as classes e objetos interagem em um cenário (BABOK, 2011).
- **Lista de Partes Interessadas, Mapas ou *Personas*:** identifica as partes interessadas que podem ser afetadas por uma iniciativa proposta ou por quem compartilha uma necessidade de negócio em comum (BABOK, 2011).
- **Mapa do Ecossistema:** é um diagrama que demonstra todos os sistemas relevantes, o relacionamento entre eles e opcionalmente objetos de dados que trafegam entre eles (PMI GUIDE, 2015).
- **Modelo de Funcionalidades:** é uma representação visual de todas as funcionalidades de uma solução organizada em uma árvore ou em uma estrutura hierárquica (PMI GUIDE, 2015).
- **Modelo de Objetivos de Negócio:** são diagramas para organização e reflexão dos objetivos, problemas de negócio, objetivos de negócio, métricas de sucesso ou funcionalidades de alto nível (PMI GUIDE, 2015).
- **Diagrama de Inter-relacionamento:** um tipo especial de diagrama de causa e efeito que retrata causas e efeitos para uma determinada

situação. Ajudam a demonstrar as causas e efeitos mais significativos envolvidos em uma determinada situação (PMI GUIDE, 2015).

- Tabela de Relatórios: é um modelo que captura em um nível detalhado todos os requisitos para um relatório (PMI GUIDE, 2015).
- Análise SWOT: é uma ferramenta valiosa para analisar rapidamente os aspectos de estado atual dos processos de negócio que estão passando por mudanças. É composta pelo agrupamento das forças, fraquezas, oportunidades e ameaças (BABOK, 2011).

Muitos estudos descrevem diversas técnicas de documentação, entretanto, algumas técnicas não são comumente utilizadas. Independente da técnica utilizada, o principal resultado da atividade de Documentação de Requisitos é o documento de requisitos, o qual é utilizado por todos os clientes, usuários finais e desenvolvedores de software. Este documento é conhecido nas organizações como “especificação funcional”, “definição de requisitos”, “especificação de requisitos de software (SRS)”, dentre outras.

Uma vez documentados os requisitos passam para a última atividade do processo denominada Validação de Requisitos.

#### **2.2.4 Validação de Requisitos**

O propósito da atividade de Validação de Requisitos é o de verificar cuidadosamente os requisitos sob o ponto de vista de consistência e completude. O objetivo desta atividade é de detectar problemas nas especificações de requisitos antes destes serem utilizados como base para o desenvolvimento do sistema (KOTONYA; SOMMERVILLE, 1998). Para (PAETSCH et al., 2003) o propósito da validação de requisitos é certificar que os requisitos possuem uma descrição aceitável do sistema que se almeja implementar.

As técnicas mais utilizadas para conduzir a validação de requisitos são Revisão de Requisitos, Testes de Requisitos (KOTONYA; SOMMERVILLE, 1998; PAETSCH et al., 2003), Prototipação e Validação de Modelo (KOTONYA; SOMMERVILLE, 1998).

A técnica de Revisão de Requisitos consiste em envolver um grupo de pessoas em uma reunião formal para ler e analisar os requisitos, procurar por problemas, se reunir para discutir os problemas e acordar um conjunto de ações necessárias para corrigir os problemas (KOTONYA; SOMMERVILLE, 1998).

A técnica de Testes de Requisitos consiste em projetar testes através da derivação de casos de testes, antes mesmo da implementação. O objetivo é que se existe alguma dificuldade em realizar esta derivação implica que existe algum tipo de problema no requisito, o qual pode ser ausência de informações no requisito ou a descrição do requisito não estar clara o suficiente para o detalhamento do que se propõe (KOTONYA; SOMMERVILLE, 1998).

A técnica de Prototipação para a atividade de validação de requisitos tem por objetivo realizar a validação dos requisitos de um sistema tomando como base os protótipos gerados na etapa de elicitacão. Como os protótipos gerados pela elicitacão não necessariamente apresentam todos os requisitos do sistema é necessário que estes protótipos evoluam durante a etapa de validacão, o que nem sempre é economicamente viável. As vantagens desta técnica para validacão é que permitem que os usuários finais realizem de maneira robusta e eficiente uma simulacão da utilizacão do sistema (KOTONYA; SOMMERVILLE, 1998).

A técnica de Validacão de Modelos consiste em analisar todos os modelos construídos (modelos de fluxos de dados das funcionalidades do sistema, modelos de objetos, modelos de eventos, modelos entidade-relacionamento, dentre outros) durante as especificacões de requisitos para analisar três objetivos: 1) demonstrar que cada modelo individualmente é auto consistente; 2) se existirem vários modelos do sistema, demonstrar que são consistentes internamente e externamente; e 3) demonstrar que estes modelos refletem os requisitos reais do sistema proposto pelas partes interessadas (KOTONYA; SOMMERVILLE, 1998). A técnica é aplicada através da conversão dos modelos de sistema para textos em linguagem natural, de maneira sistemática e com uma representacão uniformizada, que muitas vezes são gerados automaticamente por ferramentas CASE (KOTONYA; SOMMERVILLE, 1998).

Todas as atividades de Elicitacão, Análise e Negociação, Documentacão de Requisitos e Validacão de Requisitos interagem com uma atividade de suporte denominada Gerenciamento de Requisitos dentro do processo de Engenharia de Requisitos.

### **2.2.5 Gerenciamento de Requisitos**

O objetivo da atividade de Gerenciamento de Requisitos é o de capturar, documentar, disseminar, controlar alteracões, rastrear e gerenciar as informacões dos requisitos. O gerenciamento de requisitos inclui todas as atividades ligadas ao controle

de mudança e de versionamento, rastreamento e rastreabilidade entre requisitos, além do monitoramento dos estados dos requisitos (KOTONYA; SOMMERVILLE, 1998; PAESTCH et al., 2003; BATOOL et al., 2013).

Com o surgimento dos métodos ágeis a Engenharia de Requisitos sofreu adaptações para se adequar a este novo modelo de desenvolvimento de software, criando uma derivação deste processo denominada Engenharia de Requisitos Ágil.

### **2.2.6 Engenharia de Requisitos em Métodos Ágeis**

O processo tradicional de Engenharia de Requisitos enfrenta dificuldades quando o desenvolvimento de software sofre pressão por velocidade e flexibilidade, principalmente quando este desenvolvimento está sendo conduzido por meio da aplicação de abordagens ágeis.

A Engenharia de Requisitos e as abordagens ágeis parecem ser incompatíveis porque enquanto a Engenharia de Requisitos está fortemente conduzida através da documentação para compartilhar conhecimento, os métodos ágeis são focados na colaboração face à face entre usuários e desenvolvedores, para atingir os mesmos objetivos (PAESTCH et al., 2003; BATOOL et al., 2013). Os métodos ágeis são adaptativos e não preditivos como na Engenharia de Requisitos tradicional, e seus métodos foram desenvolvidos para se adaptar constantemente às mudanças. Outra característica preponderante é que são métodos orientados às pessoas, e não orientados à processos como na Engenharia de Requisitos tradicional, ou seja, confiam nas habilidades e competências das pessoas através da colaboração direta (PAESTCH et al., 2003).

Diversos estudos tentaram investigar como a Engenharia de Requisitos ocorre nas abordagens ágeis. No estudo de (CURCIO et al., 2018) foi realizada uma pesquisa para compreender quais as lacunas que existem no processo de Engenharia de Requisitos dentro do contexto do desenvolvimento ágil e quais os obstáculos que a Engenharia de Requisitos Ágil enfrenta. Neste estudo foi demonstrado que uma das principais lacunas foi negligenciar os requisitos de qualidade (uma vez que não existem práticas explícitas para o tratamento de requisitos não-funcionais nos modelos ágeis) e que as maiores dificuldades estão relacionadas com o ambiente, pessoas e recursos. Já (INAYAT et al., 2014) conduziu uma revisão sistemática da literatura e apresentou quais são as práticas adotadas pela Engenharia de Requisitos Ágil.

Alguns estudos demonstram comparativos entre as abordagens tradicionais da Engenharia de Requisitos e as abordagens ágeis. O Quadro 1 apresenta uma breve consolidação de alguns destes estudos:

**Quadro 1 - Mapeamento dos processos de Engenharia de Requisitos em abordagens ágeis de desenvolvimento, fonte (O autor, 2020)**

Atividade na Engenharia de Requisitos Tradicional	Implementação na Engenharia de Requisitos Ágil	Principais técnicas utilizadas
<b>Elicitação</b>	<ul style="list-style-type: none"> <li>• A comunicação face a face através de reuniões e a frequência de encontros auxilia a evolução dos requisitos (INAYAT et al., 2014).</li> <li>• A documentação de requisitos não-funcionais é mal definida (PAESTCH et al., 2003).</li> <li>• Os requisitos, diferentemente dos métodos tradicionais, emergem a todo tempo nos métodos ágeis (INAYAT et al., 2014).</li> </ul>	<p>(PAESTCH et al., 2003):</p> <ul style="list-style-type: none"> <li>• Entrevistas;</li> <li>• <i>Brainstorming</i>;</li> <li>• <i>Story Cards</i>; e</li> <li>• Prototipação.</li> </ul> <p>(INAYAT et al., 2014):</p> <ul style="list-style-type: none"> <li>• Entrevistas; e</li> <li>• Estórias do usuário.</li> </ul>
<b>Análise e Negociação</b>	<ul style="list-style-type: none"> <li>• <i>Product Owner</i> confirma lista de requisitos com os usuários e assegura que os requisitos estão consistentes (BATTOOL et al., 2013).</li> <li>• Priorização é realizada na lista de requisitos documentada no <i>Product Backlog</i> (BATTOOL et al., 2013)</li> </ul>	<p>(PAESTCH et al., 2003):</p> <ul style="list-style-type: none"> <li>• Técnicas de priorização por valor;</li> <li>• Sessões JAD; e</li> <li>• Modelos descartáveis.</li> </ul> <p>(INAYAT et al., 2014):</p> <ul style="list-style-type: none"> <li>• Priorização de requisitos por valor de negócio ou por risco;</li> <li>• Prototipação; e</li> <li>• <i>Goal-sketching</i>.</li> </ul>

**Quadro 1 - Mapeamento dos processos de Engenharia de Requisitos em abordagens ágeis de desenvolvimento, fonte (O autor, 2020)**

Atividade na Engenharia de Requisitos Tradicional	Implementação na Engenharia de Requisitos Ágil	Principais técnicas utilizadas
<p><b>Documentação</b></p>	<ul style="list-style-type: none"> <li>• Os Requisitos são documentados pelo <i>Product Owner</i> em um documento denominado <i>Product Backlog</i> (BATOOL et al., 2013).</li> <li>• O mínimo de documentação é realizado para eventual necessidade futura. Dúvidas são solucionadas por meio de reuniões (BATOOL et al., 2013). A documentação através das <i>User Stories</i> tem a capacidade de erradicar os desafios das constantes atualizações dos documentos de especificação de requisitos da Engenharia de Requisitos tradicional, mantendo o time sempre atualizado (INAYAT et al., 2014).</li> </ul>	<p>(PAESTCH et al., 2003):</p> <ul style="list-style-type: none"> <li>• Reuniões;</li> <li>• Modelos;</li> <li>• <i>Product Backlog</i>; e</li> <li>• Documento de requisitos simplificado.</li> </ul> <p>(INAYAT et al., 2014):</p> <ul style="list-style-type: none"> <li>• <i>User Story</i>.</li> </ul> <p>(MEDEIROS et al., 2015):</p> <ul style="list-style-type: none"> <li>• <i>Use Case</i>;</li> <li>• <i>User Story</i>;</li> <li>• <i>XXM (Extreme X-Machines)</i>;</li> <li>• <i>Feature</i>;</li> <li>• <i>Story Card</i>;</li> <li>• <i>WALL</i>;</li> <li>• <i>XSBD (eXtreme Scenario-based Design)</i>;</li> <li>• <i>Diagrama de Atividades</i>;</li> <li>• <i>Documento de Requisitos</i>;</li> <li>• <i>Task</i>;</li> <li>• <i>Wireframe</i>;</li> <li>• <i>Scenarios</i>;</li> <li>• <i>Personas</i>;</li> <li>• <i>AUC (Agile Use Case)</i>;</li> <li>• <i>ALC (Agile Loose Case)</i>;</li> <li>• <i>ACC (Agile Choose Case)</i>;</li> <li>• <i>Mind Map</i>;</li> <li>• <i>INVEST</i>;</li> <li>• <i>GPM</i>; e</li> <li>• <i>Cucumber</i>.</li> </ul>



**Quadro 1 - Mapeamento dos processos de Engenharia de Requisitos em abordagens ágeis de desenvolvimento, fonte (O autor, 2020)**

Atividade na Engenharia de Requisitos Tradicional	Implementação na Engenharia de Requisitos Ágil	Principais técnicas utilizadas
<b>Validação</b>	<ul style="list-style-type: none"> <li>A validação dos requisitos ocorre constantemente nos produtos gerados, visto que os usuários participam ativamente de todas as etapas (BATOOL et al., 2013).</li> </ul>	(PAESTCH, et al., 2003): <ul style="list-style-type: none"> <li>Testes de aceitação de <i>Story Cards</i>; e</li> <li>Reuniões de revisão.</li> </ul> (INAYAT et al., 2014): <ul style="list-style-type: none"> <li>Validação de protótipos.</li> </ul>
<b>Gerenciamento de requisitos</b>	<ul style="list-style-type: none"> <li>O gerenciamento dos requisitos ocorre durante todo o processo, contudo em cada ciclo de desenvolvimento não podem ocorrer mudanças nos requisitos já priorizados, tendo que ser ajustados no próximo ciclo (PAESTCH et al., 2003; BATOOL et al., 2013).</li> </ul>	(PAESTCH et al., 2003): <ul style="list-style-type: none"> <li><i>Product Backlog</i>;</li> <li><i>Sprints</i>; e</li> <li><i>Daily Scrums</i>.</li> </ul> (INAYAT et al., 2014): <ul style="list-style-type: none"> <li><i>Product Backlog</i>;</li> <li><i>Index cards</i>;</li> <li>Retrospectiva; e</li> <li>Planejamento contínuo.</li> </ul>

A Engenharia de Requisitos Ágil torna-se uma área emergente cujo objetivo é deixar mais flexível e conseqüentemente mais simples o processo tradicional de Engenharia de Requisitos (BATOOL et al., 2013). Entretanto, observa-se que todas as atividades da Engenharia de Requisitos Tradicional (elicitação, análise e negociação, documentação, validação e gerenciamento) ocorrem na abordagem ágil, mas com um enfoque mais iterativo e dinâmico.

Tanto na Engenharia de Requisitos tradicional quanto nas abordagens ágeis existem problemas e desafios que precisam ser superados para garantir um melhor resultado no software produzido. Algumas pesquisas já fizeram um aprofundamento neste assunto.

### 2.3 Problemas na Engenharia de Requisitos

Diversos estudos tratam dos problemas e dificuldades na execução do processo de Engenharia de Requisitos. Mesmo sendo uma área de processo amplamente conhecida e pesquisada, com o suporte de diversas técnicas de apoio, ainda assim muitos problemas na concepção dos requisitos dos projetos de software continuam ocorrendo.

Segundo (RAMINGWONG, 2012), a maioria dos problemas associados com o processo de Engenharia de Requisitos estão relacionados com as etapas de elicitação e análise de requisitos, sendo os principais problemas:

- Baixo envolvimento das partes interessadas nas etapas de elicitação e análise de requisitos;
- Uma técnica de elicitação aceita como efetiva para um projeto pode não produzir os mesmos resultados em outros;
- Muitas técnicas necessitam de orientações adicionais e flexibilidade dos praticantes para a condução do processo;
- A baixa compreensão das técnicas contribui para a seleção errônea da técnica de elicitação;
- Elevado tempo necessário para realizar a atividade de elicitação;
- Requisitos errados ou má interpretação dos requisitos;
- Mudanças rápidas nos ambientes de negócio e de tecnologia da informação forçando um menor *time-to-market*; e
- Diferentes perspectivas, percepções, preocupações, prioridades e responsabilidades das partes envolvidas durante o processo de análise e negociação dos requisitos.

O estudo conduzido por (JALIL et al., 2019) destaca como principais problemas: comunicação entre as partes interessadas, mudança de requisitos durante o projeto, linguagem humana inapropriada para termos técnicos, clientes não conseguem expressar suas necessidades de negócio, requisitos irrelevantes, baixa cooperação dos envolvidos, tempo necessário para o processo, escolha de técnicas incorretas e falha no processo de Engenharia de Requisitos.

Já (VALASKI, 2017) descreve que a deficiência na elicitação de requisitos ocorre pelo fato de que, persiste na área de Engenharia de Software, a tentativa de visualizar a tecnologia como solução de problema, sem antes focar no entendimento

do problema e na negociação de eventuais conflitos de interesses pela solução (ZANLORENCI; BURNETT, 1998).

Tais problemas sugerem a necessidade de uma nova abordagem para a Engenharia de Requisitos através de uma perspectiva que inicialmente focaria no tratamento do problema, como sugerido pelo processo de Design Thinking (LINDBERG et al., 2011; HEHN; UEBERNICKEL, 2018; SOUZA et al., 2018; VALENTIM et al., 2018).

## 2.4 Design Thinking

O processo criativo de design é considerado uma atividade humana natural e universal, que tem por objetivo criar produtos atrativos através do pensamento de ideias inovadoras (PALACIN-SILVA et al., 2017). Por um grande tempo o design tem sido utilizado como uma metodologia aplicada para transformar fisicamente produtos em algo atrativo. Ao longo dos anos observou-se que aplicar corretamente o design poderia ser o segredo de sucesso comercial em determinados produtos, desde a criação dos blocos de post-it da 3M chegando aos aparelhos eletrônicos da Apple (PALACIN-SILVA et al., 2017). Atualmente, a metodologia do design está mais voltada para desenvolver ideias que se ajustam aos desejos e necessidades dos consumidores, do que simplesmente desenvolver ideias mais atrativas para os clientes (VALENTIM et al., 2018).

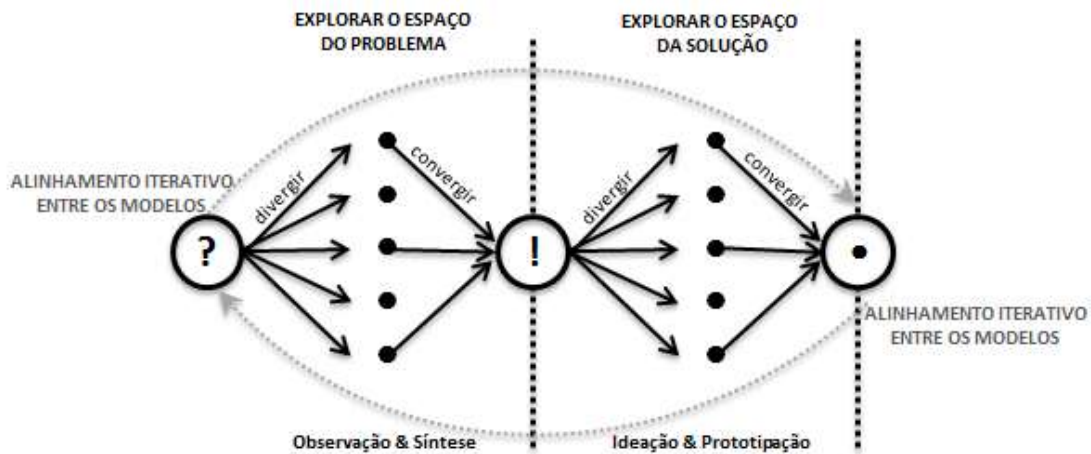
Na década de 1920 o design cresceu organicamente, deixando de ser restrito apenas à produtos, e começou a ser aplicado em diversas áreas: de estratégias de negócio e inovações sociais ao design de tecnologia. Já na década de 1969, Herbert Simon (SIMON, 1996) já havia vislumbrado o *“design como uma forma de pensamento muito além do que um processo físico”*. Como resultado deste processo, no ano 2000 as disciplinas de Design Thinking começaram a aparecer na literatura acadêmica como também nas mídias de negócio (PALACIN-SILVA et al., 2017). Na definição de (BROWN, 2008) o Design Thinking é *“trazer os princípios, as abordagens, os métodos e as ferramentas dos designers para a resolução de problemas”*.

O Design Thinking é considerado como uma metodologia para resolver problemas complexos utilizando métodos criativos (LINDBERG et al., 2011; PAULA; CORMICAN, 2016; PALACIN-SILVA et al., 2017).

Conforme demonstrado por (LINDBERG et al., 2011), diferentemente do pensamento analítico da ciência, a metodologia do pensamento do Design Thinking trata paralelamente de dois conceitos fundamentais:

- O espaço do problema e o da solução; e
- O pensamento divergente e convergente.

Estes dois conceitos foram demonstrados no conceito do diamante duplo do Design Thinking, conforme representado pela Figura 3:



**Figura 3 – Diamante duplo: espaço do problema e espaço da solução do Design Thinking, adaptado de (LINDBERG et al., 2011).**

No Design Thinking explorar o espaço do problema é explorar e observar casos de uso e cenários e sintetizar este conhecimento adquirido por pontos de vista. Já explorar o espaço da solução é testar um grande número de ideias alternativas em paralelo e elaborá-las pela aplicação de técnicas de modelagem e prototipação. O alinhamento entre os dois modelos permite manter a referência entre a solução proposta e o problema analisado garantindo uma melhor comunicação entre as equipes.

Enquanto a ciência foca seu esforço em explorar o espaço da solução, dentro de um problema já conhecido, o design explora ambos os espaços iniciando pela identificação e compreensão de problemas ocultos no espaço do problema, para na sequência explorar o espaço da solução. Esta discussão representa essencialmente o dualismo do pensamento do design (LINDBERG et al., 2011). O pensamento divergente permite explorar de maneira colaborativa todas as possibilidades, tanto no espaço do problema quanto no espaço da solução, enquanto o pensamento

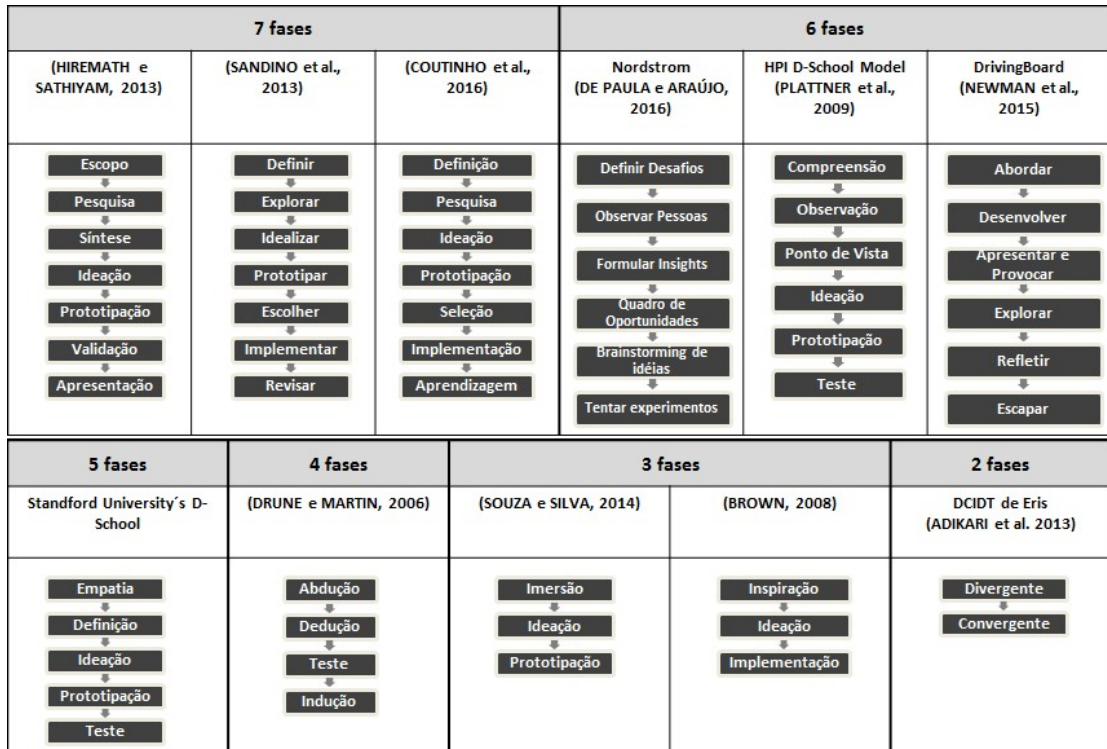
convergente é a aplicação do processo de seleção e síntese das opções mais adequadas.

O Design Thinking explora essencialmente o envolvimento de diversas partes interessadas no processo de criação, oferecendo e testando ideias rapidamente, criando engajamento de várias partes como os engenheiros de software fazem para endereçar as necessidades dos clientes (VALENTIM et al., 2018). O Design Thinking é uma metodologia para elicitare as necessidades dos clientes, ao invés de requisitos, produzindo rapidamente uma série de protótipos simples que eventualmente convergem para soluções inovadoras (VERTELLI et al., 2013). Diferentes autores propuseram implementações específicas de como conduzir o Design Thinking, muitas vezes ligadas ao contexto de aplicação, e para estas implementações denominaram-se de Modelos de Design Thinking.

#### **2.4.1 Modelos de Design Thinking**

Os modelos de Design Thinking têm como principal objetivo servir como uma orientação para a observação/síntese dentro do espaço do problema e para a ideação/prototipação dentro do espaço da solução, para problemas em diferentes contextos, através da aplicação de práticas organizadas em diferentes fases do processo.

Existem vários modelos de Design Thinking, contudo alguns destes modelos já foram aplicados à Engenharia de Software (SOUZA et al., 2017). Neste estudo os modelos de Design Thinking foram classificados conforme o número de fases de execução, conforme demonstrado pela Figura 4:



**Figura 4 - Modelos de Design Thinking, adaptado de (SOUZA et al., 2017)**

Dentre os vários modelos de Design Thinking o mais conhecido é o modelo apresentado por (BROWN, 2008), pois pode ser utilizado em diversos contextos (PAULA; CORMICAN, 2016; SOUZA et al., 2017). Para (BROWN, 2008), o Design Thinking é melhor pensado com um processo de três estágios e não uma sequência ordenada de etapas, conforme demonstrado pela Figura 5:



**Figura 5 - Três estágios do processo de Design Thinking, adaptado de (BROWN, 2008)**

O estágio de **Inspiração** propicia, através de interações com as partes envolvidas, a motivação por soluções (VALENTIM et al., 2018). Esta etapa também é conhecida como imersão, aonde são analisadas as circunstâncias (um problema, uma oportunidade ou ambos) que motivam uma busca de solução (SOUZA et al., 2018). Para (CORRAL; FRONZA, 2018) o estágio de inspiração está definido pelos seguintes passos:

- Empatizar: compreender quais as necessidades dos clientes ou dos usuários conforme seus sentimentos; e
- Definir: estabelecer as necessidades dos clientes através de um problema a ser resolvido.

O estágio de **Ideação** é o próximo estágio cujo resultado é o de gerar, desenvolver e testar uma variedade de ideias que podem conduzir à solução (VALENTIM et al., 2018). Para (CORRAL; FRONZA, 2018) o estágio de ideação é a proposição de soluções (produtos, serviços, experiências) que atendem às necessidades dos clientes.

O estágio final de **Implementação** elabora um plano para desenvolver e executar o conceito selecionado (VALENTIM et al., 2018). Esta etapa é o mapeamento de um caminho para o mercado (SOUZA et al., 2018). Para (CORRAL; FRONZA, 2018) o estágio de implementação está definido através dos seguintes passos:

- Prototipar: criar soluções que atendam um conjunto de funcionalidades solicitadas pelos usuários, permitindo que os clientes finais possam experimentá-las; e
- Testar: testar a solução mais atual, de forma a permitir que os clientes possam dar um rápido feedback e o processo possa iterar para um novo ciclo de inspiração e/ou ideação.

Os projetos ficarão em *loop* nestes três estágios, principalmente nos dois primeiros, pois as ideias refinadas em cada estágio podem conduzir à novas direções (VALENTIM et al., 2018).

Assim como ocorre na Engenharia de Requisitos os modelos de Design Thinking também se utilizam de diversas técnicas que irão suportar a execução de cada um dos estágios.

#### 2.4.2 Técnicas do Design Thinking

O Design Thinking se utiliza de inúmeras técnicas/métodos de apoio que têm por objetivo suportar o modelo durante cada um de seus estágios. No estudo de (SOUZA et al., 2017) foram mapeadas 55 técnicas/métodos disponibilizadas pelos modelos de Design Thinking e que são aplicadas na Engenharia de Software, mas deste total, apenas 9 técnicas são mais utilizadas:

- **Brainstorming:** inicia com uma pergunta provocativa com determinado tempo para que a equipe possa gerar ideias. Cada ideia é compartilhada com a equipe. Uma regra durante o processo de brainstorming é adiar o julgamento. É importante desencorajar o comportamento obstrutivo ou questionador durante este processo. Os participantes são encorajados para chegar a tantas ideias quanto possível. Isso permite que o grupo entre em um processo de agrupamento e classificação de ideias. Boas ideias naturalmente sobem ao topo, enquanto as ruins são descartadas (VALENTIM et al., 2018).
- **Persona:** uma *persona* é um arquétipo hipotético de um usuário real que descreve as metas, interesses e habilidades de um usuário. Para descrever uma *persona* é importante detalhar suas características, tais como nome, ocupação, família, amigos, sexo, idade, etnia, educação, status socioeconômico, metas, história de vida e tarefas que estas



*personas* podem executar. As *personas* podem motivar as equipes a pensarem nos usuários durante o processo de design, tornando eficientes as decisões do design, sem generalizações inapropriadas e comunicando sobre os usuários para as várias partes interessadas (VALENTIM et al., 2018).

- *Storyboard*: mostra uma série de desenhos indicando uma implementação hipotética de um serviço. Representa uma sequência de eventos, com detalhes contextuais que facilitam a análise e discussão sobre as oportunidades e problemas por meio da observação dos cenários atuais (SHEN; LO, 2018).
- *Customer Journey Map*: provê uma visualização estruturada da experiência do usuário em uma linha do tempo ou em uma sequência (SHEN; LO, 2018).
- Mapa Mental: é um esboço em que as principais categorias irradiam a partir de uma imagem central e as categorias menores são retratadas como ramos de ramos maiores (BUDD, 2004).
- Entrevista: consiste em realizar reuniões individuais ou em grupos, de maneira estruturada ou semi-estruturada com o objetivo de identificar pontos críticos nos processos, comportamentos ou nos indivíduos que precisam ser tratados durante o processo (BUDD, 2004).
- *Storytelling*: envolvem um contrato entre quem está narrando e quem está ouvindo. O papel do narrador é preparar e apresentar, através da linguagem e representação adequada, a comunicação do que se deseja narrar de maneira eficiente e eficaz através de imagens que representem a história. As estórias auxiliam a compreensão da complexidade, melhorando ou modificando percepções. As narrativas (*storytellings*) possibilitam que indivíduos possam ver a si mesmo através de uma diferente visão, para a tomada de decisões e modifiquem seu comportamento de acordo com estas novas percepções (SAMETZ; MAYDONEY, 2003).
- Serviço de *Blueprint*: neste modelo, cada aspecto individual do serviço é especificado, incorporando as perspectivas de todas as partes interessadas que possam se envolver no processo. O *Blueprint* também

especifica os detalhes de todo o processo desde o primeiro ponto de contato, bem como as interações entre diferentes partes interessadas (SHEN; LO, 2018).

- Cartões de *Insight*: representam pontos-chaves extraídos do diálogo estruturado sobre as histórias, em que os membros do grupo destacam como significativos o suficiente para serem compartilhados com outros praticantes. *Insights* são escritos como declarações completas, nomeando um ator e uma ação (LABONTE et al., 1999).

No estudo conduzido por (SOUZA et al., 2017) 15 técnicas do Design Thinking foram agrupadas e classificadas em 3 categorias:

- **Coleta, registro e organização de informações:** pesquisa exploratória, etnografia rápida, *fly on the wall*, entrevista, questionários, cartões de insight, mapa de empatia, persona e matriz de motivação;
- **Utilização em grupo:** *brainstorming* e *bodystorming*; e
- **Estimular a geração de ideias e representar possíveis soluções:** *storytelling*, *storyboards*, prototipação e *try it yourself*.

Mesmo a metodologia do Design Thinking estar ligada ao Design, já existem pesquisas científicas descrevendo a relação do Design Thinking com a Engenharia de Software.

## 2.5 Design Thinking e a Engenharia de Software

Uma das primeiras evidências da utilização do Design Thinking com a Engenharia de Software, segundo o mapeamento sistemático conduzido por (PAULA; CORMICAN, 2016), foi observada em uma publicação de 2010 em um estudo que tinha por objetivo compreender como os projetistas de software tomam decisões durante as atividades de projeto. Entretanto, nos últimos anos, percebe-se que houve um crescimento de estudos nesta área (PAULA; CORMICAN, 2016). Outros estudos demonstram que a adoção do Design Thinking no desenvolvimento de software tem crescido, pois trata-se de uma metodologia para inovação com uma abordagem centrada no usuário para pensar, dar importâncias para as necessidades humanas, avaliar a viabilidade técnica e identificar os requisitos de negócio (VALENTIM et al., 2018).

A ideia de aplicar o Design Thinking na Engenharia de Software é a de instruir um estilo de pensamento complementar, através de uma metodologia para elicitação das necessidades dos usuários, a qual expande as habilidades de resolução de problemas das equipes de desenvolvimento com o propósito de gerar resultados mais inovadores (LINDBERG et al., 2011; SOUZA et al., 2017; HEHN et al., 2020).

Escolher as técnicas mais adequadas do Design Thinking para a Engenharia de Software, possibilita o conhecimento de novas alternativas para o processo de desenvolvimento e envolve o usuário de forma mais efetiva (SOUZA et al., 2017).

Ao combinar o Design Thinking com métodos ágeis, uma vez que ambos dão ênfase nas pessoas e não em processos, parece melhorar o processo de desenvolvimento pois estes tendem a se integrar naturalmente (LINDBERG et al., 2011; PALACIN-SILVA et al., 2017). Esta combinação oferece uma metodologia consistente para executar bem dois propósitos: documentação, consistente com a Engenharia de Requisitos e gerenciamento de equipe, foco do desenvolvimento ágil (VERTELLI et al., 2013).

Entretanto, mesmo tendo aplicação em todas as fases da Engenharia de Software, percebe-se que é na Engenharia de Requisitos que o Design Thinking representa a maior contribuição. Diversos estudos apresentam que o Design Thinking pode ser aplicado como uma técnica de suporte na elicitação de requisitos (VERTELLI et al., 2013; PALACIN-SILVA et al., 2017; HEHN; UEBERNICKEL, 2018; HEHN et al., 2020), o que abre o caminho para novas investigações e pesquisas.

## **2.6 Design Thinking e a Engenharia de Requisitos**

A iteração entre o Design Thinking e a Engenharia de Requisitos já foi tratada em algumas pesquisas, como a de (HEHN et al., 2020) a qual descreve que para o Design Thinking a resolução dos problemas sempre acaba quando ocorre o entendimento e a prototipação não técnica da solução, deixando aberta a transição destes resultados para uma implementação de software. Já a Engenharia de Requisitos considera que os requisitos de software estão disponíveis e só precisam ser descobertos pela etapa de elicitação, perdendo a oportunidade de explorar melhor a compreensão dos problemas. Para fazer o uso de todo o potencial do Design Thinking, é necessário que se compreenda como ocorre sua relação com a Engenharia de Requisitos, a qual pode ocorrer por meio de três estratégias possíveis de integração (HEHN et al., 2020):

1. Início (*upfront*): rodar o Design Thinking antes de iniciar o processo de Engenharia de Requisitos;
2. Combinada (*infused*): combinar o processo de Engenharia de Requisitos com ferramentas e técnicas selecionadas do Design Thinking; ou
3. Contínua (*continuous*): combinar o uso de todos os elementos do Design Thinking e da Engenharia de Requisitos, para prover uma visão fim a fim entre as necessidades dos usuários e a solução funcional proposta.

Diversos estudos apresentam a interação entre o Design Thinking com a Engenharia de Requisitos. No estudo de caso conduzido por (HEHN; UEBERNICKEL, 2018), devido à natureza vaga do problema a ser resolvido em um projeto de exploração de plataformas digitais para o setor de energia, tomou-se a decisão de aplicar preliminarmente sessões de Design Thinking logo no início do projeto de desenvolvimento de software. O principal resultado da aplicação do Design Thinking foi uma especificação de necessidades e requisitos representados através de *mockups*, jornadas do cliente e um vídeo de *personas* interagindo com a plataforma a ser desenvolvida. As fases subsequentes de desenvolvimento, conduzidas por *sprints* através do *framework* SCRUM, detalharam um pouco mais os requisitos pela aplicação de outras ferramentas do Design Thinking como: *mood boards* (painéis semânticos), *workshops* de ideação com outros domínios de problema e *feedback capture grids*, que resultaram em épicos, histórias do usuário, fluxogramas e versões iteradas dos protótipos. A conclusão deste estudo é que o Design Thinking provê um processo estruturado para a elicitação de requisitos em casos em que os problemas são difíceis de serem resolvidos (HEHN; UEBERNICKEL, 2018). O Design Thinking pode ser utilizado como uma ferramenta complementar à Engenharia de Requisitos, mas principalmente na etapa de elicitação de requisitos (VETTERLI et al., 2013; HEHN; UEBERNICKEL, 2018; HEHN et al., 2020).

No estudo conduzido por (SOUZA; SILVA, 2015) foi apresentada a aplicação do Design Thinking focado na elicitação de requisitos para a construção de um sistema móvel para a educação. Contudo, este estudo não demonstra se o resultado das sessões de Design Thinking, principalmente protótipos, foi suficiente para se concluir o desenvolvimento da solução desenhada com efetividade e sem a necessidade de execução de outras etapas da Engenharia de Requisitos.

Já (CARROLL; RICHARDSON, 2016) propõe um *framework* de desenvolvimento para sistemas de saúde, o qual se utiliza dos princípios do Design

Thinking como um suporte para a identificação de requisitos estendendo e enriquecendo as técnicas de levantamento de requisitos tradicionais.

No entanto, o Design Thinking não é somente uma metodologia de apoio para as práticas de Engenharia de Requisitos, e sim uma forma moderna de engenharia de requisitos particularmente associada no tratamento de problemas complexos. Uma vez transformando problemas complexos em problemas mais bem definidos, estes podem na sequência ser endereçados pelas práticas bem estabelecidas da Engenharia de Requisitos (HEHN; UEERNICKEL, 2018). Algumas pesquisas exploraram que a abordagem divergente de resolução de problemas do Design Thinking é significativamente diferente das práticas mais convergentes da Engenharia de Requisitos (HEHN; UEERNICKEL, 2018), proporcionando resultados aprimorados nos processos de software quando aplicados durante as especificações.

O Quadro 2 demonstra uma compilação das análises destes estudos em relação ao contexto de exploração desta pesquisa:

**Quadro 2 – Análise dos estudos relacionados, fonte (O autor, 2020)**

Contextos de exploração desta pesquisa	(CARROLL; RICHARDSON, 2016)	(HEHN; UEERNICKEL, 2018)	(SOUZA; SILVA, 2015)	(VETTERLI et al. 2013)
Design Thinking está relacionado ao método ágil?	Não identificado no estudo	Sim, adotou SCRUM	Não identificado no estudo	Sim, o contexto sugerido é utilização de métodos ágeis
Software desenvolvido por meio da aplicação do Design Thinking era uma aplicação móvel ( <i>app</i> )?	Não identificado no estudo	Não, aplicado em software-intensive system	Sim, aplicado em mobile-learning.	Sim, o contexto sugerido é para o desenvolvimento de apps
Design Thinking auxiliou a elicitação de requisitos?	Sim, através da execução de uma etapa de pré-elicitação de requisitos para determinar as reais necessidades dos usuários (utilizando o DT) antes de conduzir a etapa de elicitação de requisitos.	Sim, requisitos de usuário e de sistema	Sim, as necessidades e os requisitos funcionais	Sim, considera que o DT oferece uma metodologia adicional para elicitação de requisitos
Design Thinking auxiliou a análise e negociação de requisitos?	Sim, a geração de protótipos pelo DT permitiu começar a análise de requisitos.	Não identificado no estudo	Não identificado no estudo	Não, o foco é o de aprendizado rápido para resolver as necessidades humanas.
Design Thinking auxiliou a documentação de requisitos?	Não identificado no estudo	Parcialmente, foram identificados: lista de funcionalidades principais, <i>mockup</i> , <i>customer journey</i> e vídeo de <i>personas</i> interagindo com plataforma	Sim, lista de requisitos funcionais.	Não identificado no estudo
Design Thinking auxiliou a validação de requisitos?	Não, a validação ocorreu através do processo tradicional da Engenharia de Requisitos.	Sim, através da validação de protótipos	Sim, através da validação de protótipos	Não identificado no estudo

Mesmo observando alguns estudos correlacionando estas duas áreas, nenhum deles teve como foco analisar sob o ponto de vista do processo de desenvolvimento como efetivamente ocorrem as interações do Design Thinking com a Engenharia de Requisitos. Conforme observado no Quadro 2 nenhum estudo cobriu, ao mesmo tempo, a elicitação, análise, documentação e validação de requisitos. Além disso, apenas dois deles trataram de aplicações móveis e também apenas dois dentre os casos estudados utilizou o método de desenvolvimento ágil.

## **2.7 Considerações sobre o capítulo**

Este capítulo apresentou a revisão da literatura relacionada à Engenharia de Requisitos e ao Design Thinking. Ele apresentou também conceitos importantes relacionados a estes dois temas bem como uma correlação entre estas duas áreas.

## CAPÍTULO 3 – MÉTODO DE PESQUISA

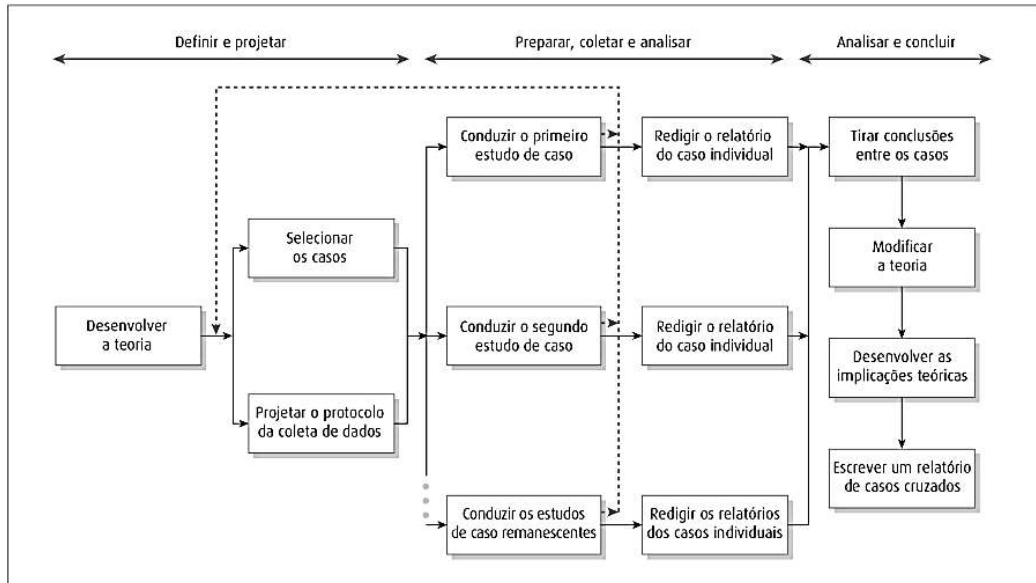
O objetivo deste capítulo é descrever o método de pesquisa que será aplicado para a compreensão da aplicação do Design Thinking na Engenharia de Requisitos em projetos de software, bem os conceitos e fundamentações metodológicas adotados para conduzir e avaliar esta pesquisa.

### 3.1 Método de Pesquisa

Esta é uma **pesquisa qualitativa e exploratória**, por meio do método de **estudo de caso (múltiplos casos)**, segundo a abordagem proposta por (YIN, 2015).

### 3.2 Estratégia de Pesquisa

A escolha do método de estudo de caso (múltiplos casos) permite ao pesquisador compreender de forma empírica e profunda como a relação do Design Thinking e da Engenharia de Requisitos acontece na prática nas organizações. Para (YIN, 2015) o projeto de pesquisa é a sequência lógica que conecta os dados empíricos às questões de pesquisa promovendo na sequência as análises, considerações e conclusões. Já (EISENHARDT, 1989) ressalta que construir teoria a partir de estudos de caso é uma estratégia de pesquisa que envolve usar um ou mais casos para criar construtos teóricos, proposições e/ou teorias intermediárias a partir de evidências empíricas. Para (YIN, 2015) o método de estudo de caso propicia uma melhor compreensão e análise de todos os dados coletados, conforme demonstrado pela Figura 6:



**Figura 6 - Processo de estudos de casos múltiplos, adaptado de (YIN, 2015).**

Para a etapa de **definir e projetar** foram realizadas as atividades de **desenvolver a teoria** por meio de uma revisão de literatura realizada nas bases científicas e a utilização de teoria de base para a elaboração dos Capítulos 1 e 2 desta dissertação. A atividade de **seleção de casos** está discriminada nas unidades de análise deste capítulo, conforme foco de exploração e alinhado com o objetivo deste estudo. Para a atividade de **de projetar o protocolo de coleta de dados** foi definido que será feito um contato prévio com as organizações objeto de estudo. Inicialmente será feita uma análise preliminar para verificar se será possível executar os estudos de caso nestas empresas, analisando se atendem aos objetivos desta pesquisa e se autorizarão a utilização dos dados nesta dissertação. Uma vez identificada a compatibilidade da organização com o objetivo da pesquisa, será firmado um Termo de Confidencialidade. Na sequência são enviadas para as organizações as informações detalhadas sobre os objetivos da pesquisa e a visão geral do escopo, de forma a possibilitar os agendamentos de datas e horários para as entrevistas. Todas as entrevistas e reuniões serão gravadas, mediante autorização prévia, para posterior análise e transcrição.

Para a etapa de **Preparar, coletar e analisar** é a fase de execução da pesquisa (a qual ocorre repetidamente para cada estudo de caso) e consiste na atividade de **conduzir o estudo de caso**, onde serão realizadas as entrevistas gravadas, dentro das unidades de análise. Na sequência é realizada a atividade **redigir o relatório do caso individual** que tem por objetivo transcrever as entrevistas e elaborar a análise



individual de cada estudo de caso validando cada uma das Proposições pela relação dos Pontos de Análise com as Proposições.

Para a etapa de **Analisar e concluir** tem por objetivo a análise e conclusão de todos os dados de campo sendo executada a atividade de tirar conclusão entre casos a qual irá produzir dados cruzados entre o vários estudos de casos realizados validando de forma cruzada cada uma das Proposições, levando em conta a relação dos Pontos de Análise e as Proposições, considerando também o nível de influência identificado pelas entrevistas. Na sequência é modificada a teoria com o registro das descobertas encontradas nos estudos de caso, desenvolve-se as implicações teóricas destas descobertas e escreve-se um relatório de casos cruzados com as conclusões finais do estudo realizado.

Pela proposta do método apresentado por (YIN, 2015) são definidos cinco componentes de pesquisa considerados essenciais no estudo de caso:

- I. Questões de pesquisa principal;
- II. Proposições;
- III. Unidades de análise;
- IV. Lógica da relação entre os dados e as proposições (conceitos de apoio às proposições);
- V. Critérios para interpretações dos dados.

Cada um destes componentes e suas definições, já endereçando os objetivos deste estudo, estão descritos detalhadamente nas próximas seções.

### 3.2.1 Questão de Pesquisa Principal

A adoção da estratégia de estudo de caso está relacionada a questões de pesquisa do tipo "como?" e "por que?" certos fenômenos acontecem (YIN, 2015). Desta forma a questão de pesquisa principal estabelecida para esta pesquisa é: ***Como o Design Thinking auxilia a Engenharia de Requisitos em projetos de software?***

### 3.2.2 Proposições

As proposições são importantes para refletir uma questão teórica e auxiliam o pesquisador a direcionar a atenção para examinar evidências relevantes no escopo do estudo de caso. Cada proposição permite a reflexão de uma questão teórica, e além de guiar o pesquisador deve ser examinada dentro do escopo do estudo. Para responder à questão de pesquisa, foram identificadas as seguintes proposições:

**P1** - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos;  
**P2** - O Design Thinking é utilizado em projetos que adotam métodos ágeis; e  
**P3** - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis.

### 3.2.3 Unidades de Análise ou Casos de Estudo

As unidades de análise descrevem o que é o caso objeto deste estudo, podendo ser representadas por um indivíduo, evento, entidade ou parte de uma organização. Neste trabalho, para as unidades de análise foram consideradas as organizações de TI do Brasil que atuam no desenvolvimento de projetos de software e que utilizam o Design Thinking para a criação de soluções de software. Serão entrevistados somente os indivíduos que participam do processo de desenvolvimento de software, como os profissionais de Design Thinking, User Experience, Arquitetos, Analistas de Sistemas e Gerentes de Desenvolvimento.

### 3.2.4 Lógica da Relação entre os Dados e as Proposições

Nesta seção estão relacionadas as principais referências da literatura para dar suporte para cada uma das proposições deste estudo. O Quadro 3 apresenta as proposições com o seu respectivo embasamento teórico:

**Quadro 3 - Conceitos de apoio para análise das proposições, fonte (O autor, 2020)**

Proposições	Conceitos de apoio à análise das proposições
P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	<p>Design Thinking provê um processo estruturado para a elicitação de requisitos para problemas complexos (HEHN; UEERNICKEL, 2018).</p> <p>Se você precisa compreender em detalhe os contextos de negócio e dos usuários, utilize o Design Thinking (HEHN et al., 2020).</p> <p>Design Thinking é consistente com as práticas iniciais de elicitação da Engenharia de Requisitos (VETTERLI et al., 2013).</p> <p>Design Thinking emprega a sensibilidade e as técnicas dos designers para corresponder às necessidades dos clientes dentro daquilo que é tecnologicamente viável (VALENTIM et al., 2018).</p> <p>Design Thinking é consistente com as primeiras práticas das atividades de elicitação de requisitos, como por exemplo, rápida prototipação e envolvimento dos interessados, como também observado pelos métodos ágeis de desenvolvimento (SOUZA; SILVA, 2015)</p>

**Quadro 3 - Conceitos de apoio para análise das proposições, fonte (O autor, 2020)**

Proposições	Conceitos de apoio à análise das proposições
P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos (continuação)	<p>Nós vislumbramos o Design Thinking como uma “arma externa” para a compreensão de problemas complexos na Engenharia de Requisitos, enquanto a Engenharia de Requisitos oferece um framework forte e integrado para o Design Thinking dentro do ciclo de desenvolvimento de software (HEHN; UEBERNICKEL, 2018).</p> <p>Nos últimos anos, diversos autores têm sido enfáticos na importância de se vislumbrar a engenharia de requisitos como um processo criativo de resolução de problemas (SOUZA; SILVA, 2015).</p> <p>Design Thinking complementa as técnicas tradicionais da engenharia de requisitos de sistemas como uma análise de requisitos dirigida por protótipos. Design Thinking vai além da elicitación de requisitos nos impedindo de ser restringido por preconceitos de soluções de software de forma isolada (CARROLL; RICHARDSON, 2016).</p> <p>(SOLEDAD et al., 2013) apresentam uma experiência em refinar requisitos através do uso do Design Thinking para um sistema de gestão do conhecimento (SOUZA; SILVA, 2015).</p> <p>Design Thinking ajuda na compreensão das necessidades reais iniciais, por exemplo tornar o problema endereçável, enquanto a Engenharia de Requisitos provê métodos para projetar a arquitetura, definir requisitos de segurança, e assim por diante (HEHN; UEBERNICKEL, 2018).</p> <p>Design Thinking provê uma metodologia para se elicitar as necessidades dos clientes e não requisitos, produzindo uma sequência rápida e simples de protótipos que eventualmente convergem para uma solução inovadora (VETTERLI et al., 2013).</p>
P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis	<p>Enquanto o Design Thinking e as práticas ágeis foram concebidas para propósitos distintos, nós podemos identificar que ambos têm por objetivo entregar soluções rápidas com um atendimento parcial de uma necessidade particular (CORRAL; FRONZA, 2018).</p> <p>Em comparação ao Design Thinking o ágil tem muitos paralelos: centrado no usuário, processo de desenvolvimento iterativo e de aprendizado e uma extensiva comunicação como time o que nos sugere que a metodologia do Design Thinking já foi introduzida para o desenvolvimento de TI (LINDBERG; MEINEL, 2011).</p> <p>Há registros de que os métodos ágeis podem combinar com o Design Thinking para aprimorar o processo de desenvolvimento. Como ambos, desenvolvimento ágil e Design Thinking, dão ênfase nas pessoas em detrimento ao processo, ambos parecem se integrar naturalmente (PALACIN-SILVA et al., 2017).</p>
P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis	<p>Como aprimorar a engenharia de requisitos para incorporar os recursos úteis das metodologias ágeis de forma a suportar o desenvolvimento de <i>apps</i>, em consonância com o desenvolvimento de grandes sistemas. Nós precisamos de um método que combina o melhor das duas abordagens: Design Thinking e Engenharia de Requisitos. (VETTERLI et al., 2013).</p>

### 3.2.5 Pontos de Análise

Os pontos de análise representam temas a serem explorados na fase de investigação, contendo os itens relevantes que não podem deixar de ser abordados durante as entrevistas, e também realizando os relacionamentos dos resultados dos mapeamentos com as proposições. Os pontos de análise, também conhecidos por questões de estudos de casos, foram propostos por (YIN, 2015) e são considerados os principais pontos do estudo de caso. A criação da estrutura de pontos de análise para coleta de dados foi descrita por (REINEHR, 2008) e é composta por:

1. **Identificação do ponto de análise:** identificador único do ponto de análise com prefixo PA seguido de um número sequencial;
2. **Descrição resumida:** – evidência que se pretende encontrar na organização;
3. **Referências teóricas:** referências que apoiam os pontos de análise;
4. **Descrição detalhada:** descreve as questões que compõem o ponto de análise e apoiam as entrevistas; e
5. **Proposições relacionadas:** proposições que estão relacionadas ao ponto de análise.

Para este trabalho foram considerados os seguintes pontos de análise, já relacionados com as Proposições, conforme apresentado no Quadro 4:

Quadro 4 - Pontos de análise das proposições, fonte (O autor, 2020)

Proposições relacionadas		
<p><b>P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos</b></p> <p><b>P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis</b></p> <p><b>P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis</b></p>		
ID do ponto de análise	Descrição resumida e detalhada	Referências teóricas
PA-01	<p><b>Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking:</b></p> <p><b>PA-01-1:</b> Qual foi o <b>objetivo</b> da organização em <b>adotar</b> o Design Thinking em projetos de software?</p> <p><b>PA-01-2:</b> Qual o <b>modelo</b> de Design Thinking adotado pela organização?</p> <p><b>PA-01-3:</b> O Design Thinking foi aplicado em quais <b>fases do processo</b> de desenvolvimento do projeto de software (Análise, Projeto, Construção, Testes)?</p> <p><b>PA-01-4:</b> O Design Thinking é considerado <b>exclusivamente</b> como uma <b>metodologia</b> para se <b>identificar os requisitos</b> necessários para uma solução de software na sua organização?</p> <p><b>PA-01-5:</b> O Design Thinking é considerado um <b>método de resolução de problemas</b> complexos com foco na solução e <b>não</b> necessariamente em <b>levantamento de requisitos</b> de software?</p> <p><b>PA-01-6:</b> O projeto evoluiu para as fases de <b>codificação</b> e <b>testes</b> logo <b>após</b> a aplicação das sessões de <b>Design Thinking</b>?</p> <p><b>PA-01-7:</b> O projeto passou por uma <b>fase de complementação de requisitos</b> logo <b>após</b> a aplicação das sessões de <b>Design Thinking</b>, antes de iniciar as fases de codificação e testes?</p> <p><b>PA-01-8:</b> Qual o <b>modelo de processo de Engenharia de Software</b> que o projeto utiliza:</p> <p style="padding-left: 40px;">a) <b>Tradicional:</b> <i>Waterfall, V-model, Spiral, UP- Unified Process</i>, outros;</p> <p style="padding-left: 40px;">b) <b>Ágil:</b> <i>SCRUM, XP-Extreme Programming, ASD- Adaptive Software Development</i>?</p> <p><b>PA-01-9:</b> Qual o <b>tipo de produto de software</b> que é desenvolvido com a utilização do Design Thinking (aplicativos móveis, software <i>desktop</i>, outros)?</p>	(LINDBERG et al., 2011; SOLEDAD et al., 2013; VETTERLI et al., 2013; SOUZA; SILVA, 2015; PALACIN-SILVA et al., 2017; CORRAL; FRONZA, 2018; HEHN; UEBERNICKEL, 2018; VALENTIM et al., 2018)

Quadro 4 - Pontos de análise das proposições, fonte (O autor, 2020)

Proposições relacionadas		
P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos		
ID do ponto de análise	Descrição resumida e detalhada	Referências teóricas
PA-02	<p><b>Verificar quais tipos de requisitos de software são identificados pelo Design Thinking:</b></p> <p><b>PA-02-1:</b> A utilização do Design Thinking no projeto de software permitiu que fossem identificados os <b>requisitos do usuário</b> (<i>serviços que a aplicação precisa prover, com suas restrições, sob o ponto de vista do usuário</i>)?</p> <p><b>PA-02-2:</b> A utilização do Design Thinking no projeto de software permitiu que fossem identificados os <b>requisitos do sistema</b> (<i>descrições detalhadas das funções do sistema, dos serviços e restrições - definição do que precisa ser desenvolvido sob o ponto de vista de arquitetura</i>)?</p> <p><b>PA-02-3:</b> <b>Requisitos funcionais</b> foram identificados nos resultados do Design Thinking (<i>declarações dos serviços que o sistema precisa prover, como o sistema precisa reagir a ações dos usuários/outros sistemas e como o sistema precisa se comportar em situações particulares</i>)?</p> <p><b>PA-02-4:</b> <b>Requisitos não-funcionais</b> foram identificados nos resultados do Design Thinking (<i>restrições nos serviços e/ou nas funções oferecidas pelo sistema como: padrões de usabilidade, tempos limite, restrições no processo de desenvolvimento, padrões de arquitetura/construção dentre outros que não necessariamente representam uma funcionalidade visível na aplicação, mas que precisam ser respeitados</i>)?</p> <p><b>PA-02-5:</b> A utilização do Design Thinking no projeto de software permitiu que fossem identificados os <b>requisitos de domínio</b> (<i>requisitos que vem do domínio do aplicativo e que refletem características desse domínio</i>)?</p>	(KOTONYA; SOMMERVILLE, 1998; HEHN et al., 2020)

Quadro 4 - Pontos de análise das proposições, fonte (O autor, 2020)

Proposições relacionadas		
P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos		
ID do ponto de análise	Descrição resumida e detalhada	Referências teóricas
PA-02	<p><b>PA-02-6:</b> A utilização do Design Thinking no projeto de software permitiu que fossem identificados requisitos <b>não-funcionais</b>?</p> <p>Exemplos:</p> <ol style="list-style-type: none"> <li>Produto → Usabilidade</li> <li>Produto → Eficiência → Desempenho</li> <li>Produto → Eficiência → Espaço</li> <li>Produto → Confiabilidade</li> <li>Produto → Portabilidade</li> <li>Organizacional → Entrega</li> <li>Organizacional → Implementação</li> <li>Organizacional → Padrões</li> <li>Externos → Interoperabilidade</li> <li>Externos → Éticos</li> <li>Externos → Legislação → Privacidade</li> <li>Externos → Legislação → Segurança</li> </ol>	(KOTONYA; SOMMERVILLE, 1998)
PA-03	<p><b>Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos:</b></p> <p><b>PA-03-1:</b> Quais foram as <b>técnicas</b> do Design Thinking que foram aplicadas no projeto de software?</p> <p><b>PA-03-2:</b> Os requisitos do projeto de software foram <b>identificados exclusivamente</b> pela aplicação do <b>Design Thinking</b>?</p> <p><b>PA-03-3:</b> Os requisitos do projeto de software foram <b>complementados</b> pela <b>Engenharia de Requisitos após</b> a aplicação do <b>Design Thinking</b>?</p>	(KOTONYA; SOMMERVILLE, 1998; BROWN, 2008; RAMINGWONG, 2012; SOLEDAD et al., 2013; SOUZA; SILVA, 2015; HEHN; UEBERNICKEL, 2018; VALENTIM et al., 2018; HEHN et al., 2020)

Quadro 4 - Pontos de análise das proposições, fonte (O autor, 2020)

Proposições relacionadas		
P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos		
ID do ponto de análise	Descrição resumida e detalhada	Referências teóricas
PA-04	<p><b>Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos:</b></p> <p><b>PA-04-1:</b> Os requisitos identificados pelo Design Thinking foram <b>analisados e negociados</b> com todas as partes interessadas?</p> <p><b>PA-04-2:</b> Foram realizadas durante o Design Thinking algum tipo de análise de <b>incompletude</b> dos requisitos?</p> <p><b>PA-04-3:</b> Foram realizadas durante o Design Thinking algum tipo de análise de <b>ambiguidade</b> dos requisitos?</p> <p><b>PA-04-4:</b> Foram realizadas durante o Design Thinking algum tipo de análise de <b>inconsistência</b> dos requisitos?</p>	(KOTONYA; SOMMERVILLE, 1998; BROWN, 2008)
PA-05	<p><b>Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos:</b></p> <p><b>PA-05-1:</b> Quais foram os <b>documentos/produtos</b> gerados pelo Design Thinking no projeto de software?</p> <p><b>PA-05-2:</b> Os requisitos identificados pelo Design Thinking foram <b>documentados</b> por algum processo formal?</p>	(KOTONYA; SOMMERVILLE, 1998; BROWN, 2008)
PA-06	<p><b>Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos:</b></p> <p><b>PA-06-1:</b> Os <b>requisitos</b> identificados pelo Design Thinking passaram por um processo de <b>validação</b> por todas as partes interessadas?</p>	(KOTONYA; SOMMERVILLE, 1998; BROWN, 2008)


### 3.2.6 Relacionamento dos Pontos de Análise com as Proposições

Como nesta pesquisa um ponto de análise está relacionado a mais de uma proposição, foi elaborado um quadro com o cruzamento de quais pontos de análise estão associados a quais proposições. O objetivo deste quadro é identificar quais pontos de análise fazem parte de uma determinada proposição, como também quais proposições são influenciadas diretamente por um ponto de análise.


A análise de cada proposição e de cada ponto de análise, para cada um dos estudos de caso abordados nesta pesquisa, se dará pela observação e análise qualitativa dos conteúdos observados nas entrevistas. Este estudo utilizará uma




categorização para determinar o nível de influência de cada ponto de análise e proposição, composto por 3 níveis, conforme definido abaixo:

-  O previsto no ponto de análise **não foi identificado** na organização.

---

-  O previsto no ponto de análise foi **identificado, mas de forma parcial ou incompleta** na organização.

---

-  O previsto no ponto de análise foi **amplamente identificado** na organização;

As categorizações dos pontos de análise serão utilizadas como suporte para determinar a decisão sobre as proposições, mas servirão para avaliar alguma eventual distorção na comparação entre as empresas, ou mesmo para chamar a atenção a algum fator que interferiu na decisão sobre a proposição, mesmo com os indicadores apontando para outra direção.

Quando na interseção entre a proposição e o ponto de análise tiver a indicação do sinal “◆” significa que há relacionamento direto entre a proposição e o respectivo ponto de análise. O mapeamento resultante está representado no Quadro 5, destacando o ponto de interseção para cada uma das proposições e pontos de análise:

Quadro 5 - Intersecção entre pontos de análise e proposições, fonte (O autor, 2020)

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-01</b> <b>Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking</b>	PA-01-1: Qual foi o objetivo da organização em adotar o Design Thinking em projetos de software?	◆	◆	◆
	PA-01-2: Qual o modelo de Design Thinking adotado pela organização?	◆	◆	◆
	PA-01-3: O Design Thinking foi aplicado em quais fases do processo de desenvolvimento do projeto de software (Análise, Projeto, Construção, Testes)?	◆		
	PA-01-4: O Design Thinking é considerado exclusivamente como uma metodologia para se identificar os requisitos necessários para uma solução de software na sua organização?	◆		
	PA-01-5: O Design Thinking é considerado um método de resolução de problemas complexos com foco na solução e não necessariamente em levantamento de requisitos de software?	◆		
	PA-01-6: O projeto evoluiu para as fases de codificação e testes logo após a aplicação das sessões de Design Thinking?	◆		
	PA-01-7: O projeto passou por uma fase de complementação de requisitos logo após a aplicação das sessões de Design Thinking, antes de iniciar as fases de codificação e testes?	◆		
	PA-01-8: Qual o modelo de processo de Engenharia de Software que o projeto utiliza: a) Tradicional: <i>Waterfall, V-model, Spiral, UP-Unified Process</i> , outros; b) Ágil: <i>SCRUM, XP-Extreme Programming, ASD-Adaptive Software Development</i> ?			◆
	PA-01-9: Qual o tipo de produto de software que é desenvolvido com a utilização do Design Thinking (aplicativos móveis, software <i>desktop</i> , outros)?			◆

**Quadro 5 - Intersecção entre pontos de análise e proposições, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-02</b> Verificar quais tipos de requisitos de software são identificados pelo Design Thinking	PA-02-1: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do usuário (serviços que a aplicação precisa prover, com suas restrições, sob o ponto de vista do usuário)?	◆		
	PA-02-2: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do sistema (descrições detalhadas das funções do sistema, dos serviços e restrições - definição do que precisa ser desenvolvido sob o ponto de vista de arquitetura)?	◆		
	PA-02-3: Requisitos funcionais foram identificados nos resultados do Design Thinking (declarações dos serviços que o sistema precisa prover, como o sistema precisa reagir a ações dos usuários/outros sistemas e como o sistema precisa se comportar em situações particulares)?	◆		
	PA-02-4: Requisitos não-funcionais foram identificados nos resultados do Design Thinking (restrições nos serviços e/ou nas funções oferecidas pelo sistema como: padrões de usabilidade, tempos limite, restrições no processo de desenvolvimento, padrões de arquitetura/construção dentre outros que não necessariamente representam uma funcionalidade visível na aplicação, mas que precisam ser respeitados)?	◆		
	PA-02-5: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos de domínio (requisitos que vem do domínio do aplicativo e que refletem características desse domínio)?	◆		
	PA-02-6: A utilização do Design Thinking no projeto de software permitiu que fossem identificados requisitos não-funcionais?	◆		
<b>PA-03</b> Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos	PA-03-1: Quais foram as técnicas do Design Thinking que foram aplicadas no projeto de software?	◆		
	PA-03-2: Os requisitos do projeto de software foram identificados exclusivamente pela aplicação do Design Thinking?	◆		
	PA-03-3: Os requisitos do projeto de software foram complementados pela Engenharia de Requisitos após a aplicação do Design Thinking?	◆		

**Quadro 5 - Intersecção entre pontos de análise e proposições, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-04</b> Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos:	PA-04-1: Os requisitos identificados pelo Design Thinking foram analisados e negociados com todas as partes interessadas?	◆		
	PA-04-2: Foram realizadas durante o Design Thinking algum tipo de análise de incompletude dos requisitos?	◆		
	PA-04-3: Foram realizadas durante o Design Thinking algum tipo de análise de ambiguidade dos requisitos?	◆		
	PA-04-4: Foram realizadas durante o Design Thinking algum tipo de análise de inconsistência dos requisitos?	◆		
<b>PA-05</b> Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos	PA-05-1: Quais foram os documentos/produtos gerados pelo Design Thinking no projeto de software?	◆		
	PA-05-2: Os requisitos identificados pelo Design Thinking foram documentados por algum processo formal?	◆		
<b>PA-06</b> Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos	PA-06-1: Os requisitos identificados pelo Design Thinking passaram por um processo de validação por todas as partes interessadas?	◆		

### 3.2.7 Critérios para Interpretações dos Dados

A análise dos resultados será guiada através dos Pontos de Análise definidos neste estudo. Os Pontos de Análise servirão como o guia tanto para a análise individual de cada caso como também para a análise cruzada de todos os estudos de caso executados. Os dados coletados durante as entrevistas serão transcritos e analisados utilizando a matriz de relação entre os Pontos de Análise e as Proposições (conforme descrito no Capítulo 3.2.6), também levando em consideração os níveis de influência que a organização analisada apresentou para cada uma destas intersecções.

### 3.2.8 Protocolo de Pesquisa

O protocolo de pesquisa constitui a organização e a estrutura básica de apoio à pesquisa. Este protocolo é composto por:

1. Termo de Confidencialidade;
2. Apresentação da Pesquisa;
3. Visão Geral da Pesquisa;
4. Procedimentos Operacionais; e
5. Pontos de Análise.

O documento de **Termo de Confidencialidade**, no APÊNDICE A, é o documento que garante o sigilo das informações fornecidas ao entrevistador. Esse documento é entregue ao responsável pelo pesquisador dentro da organização no dia da visita.

O documento de **Apresentação da Pesquisa**, no APÊNDICE B, é o documento que tem por objetivo fornecer uma visão consolidada e geral da pesquisa para a organização.

O documento de **Visão Geral da Pesquisa**, no APÊNDICE C, é o documento que tem por objetivo fornecer a identificação, objetivos, a questão a ser pesquisada, público-alvo, procedimentos operacionais, confidencialidade das informações, papéis do público-alvo, questões e proposições.

Os **Procedimentos Operacionais** têm como objetivo apoiar o pesquisador com relação aos passos que devem ser seguidos em cada nas organizações para a realização do estudo de caso. Este procedimento passa pelo contato prévio com as organizações alvo e na sequência o envio de informações mais detalhadas, como os documentos de **Apresentação da Pesquisa** e **Visão Geral da Pesquisa**, e então organizar o planejamento das datas e horários das reuniões presenciais para a condução dos estudos de caso. Todas as reuniões serão gravadas, conforme autorização dos participantes, para posterior transcrição e análise. Após a análise do relatório resultante das entrevistas, se for identificada alguma dúvida é agendada uma nova reunião presencial com a organização com o objetivo de complementar alguma lacuna ou sanar eventuais dúvidas que tenham surgido durante entrevistas. Uma vez concluído o ciclo das entrevistas, é elaborada a análise individual de cada estudo de caso e posteriormente é elaborada a análise consolidada de todos os estudos de caso.

Os **Pontos de Análise**, descritos anteriormente, têm como objetivo direcionar o foco do investigador durante a fase de entrevistas.

### **3.3 Considerações sobre o capítulo**

Neste capítulo foi explanado qual o método de pesquisa foi selecionado para esta dissertação de forma a responder aos objetivos de pesquisa presentes no Capítulo 1.2. O propósito é demonstrar como será organizada e executada a pesquisa cujos resultados estarão documentados nos próximos capítulos deste documento.

## CAPÍTULO 4 - ESTUDOS DE CASO

Este capítulo descreve os estudos de caso realizados em três organizações, bem como as respectivas análises. Ao final de cada estudo de caso foi realizada uma discussão dos resultados, de forma individualizada, face à cada uma das proposições.

Para preservar a confidencialidade das informações aqui descritas, as unidades de análise serão referenciadas por letras e organizadas em ordem alfabética e não representando a ordem de execução dos estudos de caso.

### 4.1 Unidades de análise estudadas

As organizações estudadas seguiram os critérios definidos no Capítulo 3 desta dissertação. O Quadro 6 descreve o perfil das organizações analisadas nos estudos de caso.

**Quadro 6 – Perfil das unidades de análise e dos entrevistados, fonte (O autor, 2020)**

<b>Organização</b>	<b>Porte</b>	<b>Capital</b>	<b>Perfil entrevistado(s)</b>
A	Grande	Privado	Gerente de user experience Gerente de desenvolvimento de software
B	Grande	Privado	Gerente de desenvolvimento de software
C	Grande	Aberto	Gerente de user experience

Na Organização A, o participante com o perfil de gerência de user experience trabalha na empresa há mais de três anos, com grande experiência na utilização de métodos criativos como o Design Thinking e é responsável por conduzir o processo de criação de soluções de softwares inovadoras para diversas indústrias da iniciativa privada. Já o participante com o perfil de gerente de desenvolvimento de software atua na coordenação das atividades de desenvolvimento de software.

Na Organização B, o participante trabalha na empresa há mais de três anos, com vasto conhecimento na combinação do Design Thinking com o desenvolvimento de software em projetos de diversas indústrias, dentro de setores do governo como também em setores da iniciativa privada.

Na Organização C, o participante trabalha na empresa há mais de treze anos tendo sido o responsável pela combinação de métodos criativos, como o Design Thinking com o desenvolvimento de software e com um vasto conhecimento desta combinação dentro do contexto da engenharia de requisitos.

## **4.2 Descrição da Coleta de Dados e Tratamento dos Dados**

A coleta de dados foi realizada por meio de entrevistas semi-estruturadas seguindo os pontos de análise previstos no Quadro 4 do Capítulo 3.2.5. As entrevistas foram gravadas e posteriormente transcritas integralmente para a análise dos dados, conforme o protocolo definido no Capítulo 3.2.

Na Organização A a entrevista foi realizada na sede da empresa em Curitiba, por haver disponibilidade de todos os integrantes para se realizar presencialmente. Já na Organizações B as entrevistas foram realizadas por videoconferência, através da plataforma do Zoom, pelo fato da pessoa entrevistada estar localizada em uma unidade de negócio da organização em Recife. Já para a Organização C, estava prevista a realização das entrevistas presencialmente em uma das unidades da organização em São Paulo, mas devido à quarentena imposta pela pandemia de COVID-19, as entrevistas acabaram sendo realizadas por videoconferência, através da plataforma do Zoom. Em todas as organizações, todos os participantes permitiram que os áudios das entrevistas fossem gravados.

Com a finalização da transcrição dos áudios gravados, iniciaram-se as análises das proposições, tomando como base as informações presentes nas transcrições.

## **4.3 Organização A**

### **4.3.1 Caracterização da Organização A**

A Organização A é uma empresa de grande porte com capital privado e vem atuando no mercado de desenvolvimento de software há mais de 20 anos. Seu principal foco está voltado em soluções para o setor de telecomunicações, mas também tendo forte investimento em outros setores como varejo, educação, saúde, indústria, dentre outros.

Esta organização é uma das empresas pioneiras na adoção de métodos ágeis para o desenvolvimento de software e possui uma equipe com mais de 200



profissionais distribuídos em três centros de desenvolvimento no Brasil e um centro de desenvolvimento nos EUA.

#### **4.3.2 Contextualização da Adoção do Design Thinking com a Engenharia de Requisitos na Organização A**

Os processos de desenvolvimento de software desta organização, os quais vinham evoluindo há mais de 20 anos, começaram a incorporar há aproximadamente 3 anos a metodologia do pensamento do design, cuja base é o Design Thinking. Diversas adaptações foram realizadas na metodologia básica do Design Thinking para que fosse possível combinar esta metodologia com o processo de desenvolvimento existente na empresa.

Esta organização ressalta que, quando é possível aplicar o Design Thinking no início dos projetos ou antes mesmo dos projetos serem contratados pelos clientes, existe uma contribuição da aplicação desta metodologia para o levantamento de requisitos de software, uma vez que a aplicação do Design Thinking em uma etapa de concepção do produto permite realizar um aprofundamento, com todos os envolvidos, no entendimento de quais problemas o software precisa tratar.

A combinação da metodologia do Design Thinking com os processos de engenharia de software, nesta organização, visa compreender dois pontos fundamentais:

- As necessidades do cliente, muitas vezes por meio de um aprofundamento de quais os problemas estes clientes, ou os clientes dos clientes, têm enfrentado; e
- O que é tecnologicamente viável, dentro da complexidade tecnológica do cliente, para a solução destes problemas através de uma abordagem de desenvolvimento de software.

O modelo de Design Thinking adotado é baseado no modelo de Tim Brown (BROWN, 2008) porém, a organização informou que realizou algumas adaptações neste modelo, dividindo-o em mais etapas para uma melhor adequação aos processos já existentes.

#### **4.3.3 Resultados das Análises das Proposições na Organização A**

## **P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos:**

A organização começou a adotar o Design Thinking em um produto próprio comercializado no mercado, mas com um foco muito voltado à geração de telas. Com o passar do tempo, a empresa começou a prestar serviços de desenvolvimento de software fora do contexto deste produto e tomou a decisão de utilizar amplamente o Design Thinking para agregar mais valor aos usuários.

Ao combinar a aplicação do Design Thinking com a Engenharia de Requisitos a organização conseguiu aprimorar o processo de desenvolvimento, pois garantiu uma maior proximidade com o usuário final do produto, seja ele o cliente contratante ou o cliente deste cliente. Esta abordagem permitiu não somente uma maior assertividade na **elicitação de requisitos** de usuário e requisitos de sistema, como também uma maior assertividade no atingimento dos objetivos de negócio:

*“[...] A gente entende, de acordo com os objetivos do negócio, o que faz realmente sentido ser implementado [...] (p.13)”*

*“[...] a gente fez um sistema onde a gente foi até o cliente final do nosso cliente, ver como que era o dia a dia dele. Aí eles já tinham uma plataforma, então a gente entendeu como era essa plataforma, o quanto (p.12) ela ajudava ou atrapalhava a vida do cliente final [...] (p.13)”*

*“[...] principalmente, trabalhar com as limitações tecnológicas, para que, também, a gente não eleve a expectativa do cliente, na hora de ficar idealizando o que vai ser feito, fazendo um protótipo, e aí na hora do desenvolvimento metade daquilo não dá para ser feito [...] (p.7)”*

A combinação do Design Thinking com as etapas de **elicitação de requisitos** garante uma maior clareza no escopo, pois as equipes de desenvolvimento atuam em conjunto com as equipes de design durante a realização das sessões de Design Thinking. As equipes do design têm o foco voltado para a elicitação de requisitos de usuário, mas realizam também algum tipo de elicitação de requisitos de sistema do tipo funcionais, como restrições tecnológicas e principalmente requisitos de usabilidade. Já as equipes de desenvolvimento têm o foco voltado para a elicitação de requisitos de sistema do tipo funcionais e não-funcionais durante este processo.

A proposta de envolvimento das equipes do design com as equipes de desenvolvimento de software, durante a etapa de **elicitação de requisitos**, é justamente para garantir que o que será entregue pelos projetos de software atenderá as necessidades de negócio do cliente. Muitas vezes esta interação não se restringe

à fase de Engenharia de Requisitos e se estende para outras fases do processo de Engenharia de Software, conforme observado:

*“[...] Porque a gente tem um desafio aqui dentro, ainda muito grande, que a gente não quer atuar só até a (prototipação), a gente quer continuar, justamente para validar o que está sendo entregue para o cliente, a ponto de: será que o que eu fiz de protótipo é o que está sendo entregue? Ou: nesse caminho, não tem alguma coisa que eu preciso alterar? Ou que mudou alguma coisa dentro do cliente, alguma prioridade, e eu tenho que alterar aqui? Então é para que a gente faça essa validação pós-entrega, se isso está mesmo fazendo sentido para a vida, para o dia a dia do cliente [...] (p.7)”*

Também foi observado que algum tipo de **análise e negociação de requisitos** ocorre dentro das sessões de Design Thinking, principalmente quando as equipes de design e de desenvolvimento apresentam os protótipos funcionais navegáveis ao cliente. Durante este processo o cliente revisa se todas as funcionalidades do produto e todas as restrições de usabilidade foram endereçadas. Como normalmente as equipes de desenvolvimento trabalham junto com as equipes de design, existe uma negociação interna sobre os requisitos identificados pelo Design Thinking, principalmente para garantir que a proposta de solução seja tecnicamente viável de ser implementada, e se necessário alguns ajustes são realizados nos requisitos:

*“[...] a gente entregou um protótipo, aquilo não está escrito na pedra, então se você encontrar algum problema, ou você está vendo que está muito complexo em desenvolver, que poderia ser de um jeito, de repente uma outra solução, a gente sempre pede para que eles nos acionem, e a gente tenha esse processo mais evolutivo. Até para não chegar lá na hora do teste, e aí sim a gente ver que foi feito diferente [...] (p.14)”*

Durante as etapas de descobertas do Design Thinking, ocorre também algum tipo de **documentação de requisitos**, seja por meio de documentos de especificação de produtos, personas, jornadas do usuário e principalmente pela elaboração dos protótipos funcionais navegáveis (telas de sistema com botões, menus e links que simulam a transição das telas em uma condição muito próxima ao software que será gerado pelo desenvolvimento) o qual é o principal sub-produto resultante destas sessões. Normalmente as equipes de desenvolvimento de software trabalham em conjunto com as equipes de design na elaboração destas documentações:

*“[...] A gente gera o protótipo navegável. A gente tem toda a parte de documentação dessa inception, dessa parte de descoberta, e o dev também participa junto, e no final, tanto o dev quanto a parte de design, a gente cria uma especificação do produto [...] (p.11)”*

Em casos específicos, quando o cliente exige um nível de documentação maior, o time de desenvolvimento desenvolve documentos complementares, muitas vezes ligado ao processo de análise da metodologia adotada pelo cliente. Eventualmente, o time de design também auxilia o time de desenvolvimento na geração de documentações complementares, neste caso as documentações relacionadas às especificações das telas do sistema.

Por fim, observou-se que as sessões de Design Thinking auxiliam a etapa de **validação de requisitos**, onde os requisitos produzidos pelas sessões de Design Thinking e materializados nas documentações, principalmente protótipos funcionais navegáveis, são validados pelo cliente para garantir que estes estejam corretamente endereçados e atendam às suas necessidades:

“[...] Pesquisador: *Então vocês fazem uma validação, mesmo, dos requisitos, dentro do protótipo que foi gerado, dos produtos que foram gerados, com o cliente?*

Organização A: *Sim.*

Pesquisador: *Usando o próprio Design Thinking?*

Organização A: *Sim, exatamente [...] (p.8)*”.

Contudo, a organização afirma que nem sempre o usuário final do produto faz a validação dos protótipos, mas que esta validação sempre ocorre com o responsável pelo projeto no cliente.

Para esta organização as fases de Design, Construção e Testes do processo de Engenharia de Software já iniciam suas atividades logo após a finalização das atividades do Design Thinking, sem a necessidade de execução de uma fase formal de Engenharia de Requisitos, a qual normalmente ocorre concomitantemente com as sessões de Design Thinking. Isso é possível devido à proximidade da equipe de desenvolvimento com a equipe de design, desde o início do projeto, para o entendimento, construção e validação das propostas de soluções junto aos clientes, e que raramente é necessária alguma complementação de requisitos:

“[...] às vezes é porque passou alguma coisa despercebida, porque acontece, mas, por padrão, não. Não é um fluxo padrão. Normalmente quando chega para o dev desenvolver a parte de front já está mais redondo [...] (p.11)”.

Uma vez que os times de construção iniciam a codificação do software as equipes de design já iniciam o próximo ciclo de Design Thinking, até que todos os problemas que precisam ser tratados pelo software sejam compreendidos e os requisitos especificados.

## **P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis:**

Nesta organização a aplicação do Design Thinking **não está ligada** ao **modelo de processo de desenvolvimento** adotado. A escolha da utilização do Design Thinking está intimamente ligada à característica do projeto e da problemática que precisa ser resolvida. Muitas vezes, a escolha de um modelo de processos de desenvolvimento é imposta pelo cliente. Existem projetos que aplicam o Design Thinking em um processo de desenvolvimento que adota o modelo ágil, mas existem projetos que aplicam o Design Thinking em projetos que adotam o modelo cascata:

*“ [...] Depende muito do cliente. A empresa preza pelo ágil, então a gente tenta fazer um processo ágil, o mais voltado para a entrega contínua [...] (p.8)”,*

*“ [...] se o cliente quiser trabalhar com o modelo waterfall, a gente trabalha com o modelo waterfall [...] (p.8)”.*

Quando a organização pode sugerir um processo de desenvolvimento, o **modelo ágil torna-se o modelo preferencial**, sendo o predominante na organização:

*“ [...] A gente trabalha com muita empresa que está se adequando e tem interesse em trabalhar com modelos ágeis. Então o modelo ágil predomina, de uma forma geral [...] (p.8)”.*

A organização tem preferência pela adoção do Design Thinking com o método ágil, porque os ciclos de aprofundamento de entendimento do problema oriundos do design geram mais sinergia com os métodos ágeis por meio de seus ciclos mais curtos de entregas, o que garante uma maior assertividade e maior velocidade no atendimento das expectativas do cliente. Normalmente, as equipes de design vão liberando entregas para as equipes de desenvolvimento e enquanto estes times constroem o software, as equipes de design conseguem avançar em novas ondas de descobertas com o cliente. O ciclo de entregas do ágil permite ao cliente experimentar o produto e repriorizar as frentes de Design Thinking conforme as soluções de software são entregues.

## **P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis:**

A organização descreve que a aplicação do Design Thinking é mais utilizada em softwares desktops, mas não por uma característica específica de melhor adequação na aplicação do Design Thinking com esta família de software e sim pelo predomínio de demanda desta família de software pelos clientes. Entretanto, a

empresa afirma que houve um crescimento da aplicação desta metodologia em aplicações móveis, principalmente em grandes empresas:

“[...] Pesquisador: *A família de software que normalmente vocês desenvolvem, são softwares desktop, softwares móveis, plataformas, ou o que?*

Organização A: *A maioria desktop, ainda. Mas a gente já está cada vez crescendo mais em móvel [...] (p.9)*”.

“[...] *a gente tem tido muitos clientes que estão procurando. Há três anos atrás, quando eu entrei, não tinha nenhum projeto de aplicativo, nada. Hoje a gente já tem quatro, cinco. E são grandes projetos, para empresas grandes [...] (p.9)*”.

O Quadro 7 demonstra o resumo dos resultados obtidos e analisados no estudo de caso da Organização A:














**Quadro 7 – Resultados obtidos no estudo de caso da Organização A, fonte (O autor, 2020)**

<b>P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos</b>			
<b>Elicitação de Requisitos</b>	<b>Análise e Negociação de Requisitos</b>	<b>Documentação de Requisitos</b>	<b>Validação de Requisitos</b>
Foi observado que as atividades de elicitación de requisitos ocorre com a presença das equipes de design e de desenvolvimento durante a execução das sessões de Design Thinking.	A análise e negociação de requisitos se dá em dois momentos durante as sessões de Design Thinking: 1) Entre as equipes de design e de desenvolvimento com a finalidade de garantir a viabilidade tecnológica da solução; e 2) Com o cliente para alinhamento do escopo da solução.	A documentação de requisitos é realizada pelas equipes de design e pelas equipes de desenvolvimento, durante as sessões de Design Thinking.	A validação de requisitos com o usuário do projeto ocorre ao término das dinâmicas de Design Thinking.
<b>Tipos de requisitos identificados pelo Design Thinking</b>	Requisitos de usuário e requisitos de sistema dos tipos funcionais e alguns não-funcionais, mas não foi observado a derivação de requisitos de sistema do tipo de domínio.		
<b>Técnicas utilizadas pelo Design Thinking</b>	Documentos de especificação de produtos, personas, jornadas do usuário e protótipos.		
<b>P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis</b>			
<b>Modelos de desenvolvimento de software que utilizam o Design Thinking</b>	Predominantemente no modelo ágil, contudo também foi observado a aplicação no modelo waterfall. A escolha do modelo é uma decisão dos clientes.		
<b>P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis</b>			
<b>Famílias de software que utilizam o Design Thinking</b>	A maioria dos softwares desenvolvidos com esta abordagem são softwares desktops, entretanto a organização afirma ter observado um crescimento de demanda para sistemas móveis nos últimos anos.		










#### 4.3.3.1 Análise dos níveis de influência nos pontos de análise e proposições

O Quadro 8 demonstra o nível de influência identificado em cada um dos pontos de análise e as proposições no estudo de caso da Organização A:

**Quadro 8 – Nível de influência entre pontos de análise e proposições na Organização A, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-01</b> <b>Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking</b>	PA-01-1: Qual foi o objetivo da organização em adotar o Design Thinking em projetos de software?			
	PA-01-2: Qual o modelo de Design Thinking adotado pela organização?			
	PA-01-3: O Design Thinking foi aplicado em quais fases do processo de desenvolvimento do projeto de software (Análise, Projeto, Construção, Testes)?			
	PA-01-4: O Design Thinking é considerado exclusivamente como uma metodologia para se identificar os requisitos necessários para uma solução de software na sua organização?			
	PA-01-5: O Design Thinking é considerado um método de resolução de problemas complexos com foco na solução e não necessariamente em levantamento de requisitos de software?			
	PA-01-6: O projeto evoluiu para as fases de codificação e testes logo após a aplicação das sessões de Design Thinking?			
	PA-01-7: O projeto passou por uma fase de complementação de requisitos logo após a aplicação das sessões de Design Thinking, antes de iniciar as fases de codificação e testes?			
	PA-01-8: Qual o modelo de processo de Engenharia de Software que o projeto utiliza: a) Tradicional: <i>Waterfall, V-model, Spiral, UP-Unified Process</i> , outros; b) Ágil: <i>SCRUM, XP-Extreme Programming, ASD-Adaptive Software Development</i> ?			
	PA-01-9: Qual o tipo de produto de software que é desenvolvido com a utilização do Design Thinking (aplicativos móveis, software <i>desktop</i> , outros)?			

**Quadro 8 – Nível de influência entre pontos de análise e proposições na Organização A, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-02</b> Verificar quais tipos de requisitos de software são identificados pelo Design Thinking	PA-02-1: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do usuário (serviços que a aplicação precisa prover, com suas restrições, sob o ponto de vista do usuário)?			
	PA-02-2: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do sistema (descrições detalhadas das funções do sistema, dos serviços e restrições - definição do que precisa ser desenvolvido sob o ponto de vista de arquitetura)?			
	PA-02-3: Requisitos funcionais foram identificados nos resultados do Design Thinking (declarações dos serviços que o sistema precisa prover, como o sistema precisa reagir a ações dos usuários/outros sistemas e como o sistema precisa se comportar em situações particulares)?			
	PA-02-4: Requisitos não-funcionais foram identificados nos resultados do Design Thinking (restrições nos serviços e/ou nas funções oferecidas pelo sistema como: padrões de usabilidade, tempos limite, restrições no processo de desenvolvimento, padrões de arquitetura/construção dentre outros que não necessariamente representam uma funcionalidade visível na aplicação, mas que precisam ser respeitados)?			
	PA-02-5: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos de domínio (requisitos que vem do domínio do aplicativo e que refletem características desse domínio)?			
	PA-02-6: A utilização do Design Thinking no projeto de software permitiu que fossem identificados requisitos não-funcionais ?			
<b>PA-03</b> Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos	PA-03-1: Quais foram as técnicas do Design Thinking que foram aplicadas no projeto de software?			
	PA-03-2: Os requisitos do projeto de software foram identificados exclusivamente pela aplicação do Design Thinking?			
	PA-03-3: Os requisitos do projeto de software foram complementados pela Engenharia de Requisitos após a aplicação do Design Thinking?			



**Quadro 8 – Nível de influência entre pontos de análise e proposições na Organização A, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-04</b> Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos:	PA-04-1: Os requisitos identificados pelo Design Thinking foram analisados e negociados com todas as partes interessadas?	☹️		
	PA-04-2: Foram realizadas durante o Design Thinking algum tipo de análise de incompletude dos requisitos?	😊		
	PA-04-3: Foram realizadas durante o Design Thinking algum tipo de análise de ambiguidade dos requisitos?	☹️		
	PA-04-4: Foram realizadas durante o Design Thinking algum tipo de análise de inconsistência dos requisitos?	😊		
<b>PA-05</b> Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos	PA-05-1: Quais foram os documentos/produtos gerados pelo Design Thinking no projeto de software?	😊		
	PA-05-2: Os requisitos identificados pelo Design Thinking foram documentados por algum processo formal?	😊		
<b>PA-06</b> Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos	PA-06-1: Os requisitos identificados pelo Design Thinking passaram por um processo de validação por todas as partes interessadas?	😊		

## 4.4 Organização B

### 4.4.1 Caracterização da Organização B

A Organização B é uma empresa brasileira de grande porte, com capital privado, e vem atuando em áreas de tecnologia de informação e de desenvolvimento de software há mais de 31 anos, atendendo diversos setores como educação, saúde, finanças, telecom, governo, varejo, indústria e marketing.

Esta organização possui operações em vários estados brasileiros e com presença internacional em diversos países contando com uma equipe de mais de 25.000 colaboradores. Possui o foco de atuação muito voltado para a análise de negócios com uma extensa visão de mercado, criando soluções personalizadas de

transformação digital para seus clientes. A organização tem uma estratégia muito voltada a soluções inovadoras por meio de serviços que vão desde a aplicação de automação, inteligência artificial, cloud, internet das coisas (IoT) e user experience (UX) até atividades de consultoria, marketing, desenvolvimento de software, outsourcing e service desk.

#### **4.4.2 Contextualização da Adoção do Design Thinking com a Engenharia de Requisitos na Organização B**

A aplicação do Design Thinking com o desenvolvimento de software nesta organização, especificamente na área em que foi conduzido este estudo de caso, foi motivado pela necessidade da Organização B propor uma nova abordagem de modelagem de negócio para um cliente do setor financeiro. O cliente tinha processos burocráticos e os produtos não conseguiam ser desenvolvidos, assim surgiu a necessidade de se obter formas efetivas de facilitação para criação de novos produtos por meio do desenvolvimento de software. Muitas vezes a concepção de um novo produto exigia o tratamento de problemas complexos, e o correto entendimento e tratamento destes problemas fez com que técnicas de Design Thinking fossem a abordagem mais adequada para estas situações:

*“[...] ensinar uma nova abordagem de entendimento e modelagem do negócio para o banco [...] (p.4)”.*

A utilização do Design Thinking, além de promover o entendimento do negócio, propiciou a criação de um novo produto para este cliente, o qual estava sendo desenvolvido pela equipe de projeto nesta organização.

A organização não explicitou se adota um modelo específico de Design Thinking neste cliente, mas relatou que faz a combinação de diversas técnicas de Design Thinking para garantir a efetividade nos resultados dos projetos.

#### **4.4.3 Resultados das Análises das Proposições na Organização B**

##### **P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos:**

Foi observado que o Design Thinking neste estudo de caso, além de ser uma metodologia aplicada à compreensão dos fluxos de negócio e de apoio para a criação de novos produtos, estava relacionado dentro do contexto da Engenharia de Software exclusivamente com o processo de Engenharia de Requisitos.

A combinação do Design Thinking com o desenvolvimento de software permitiu que os requisitos de software fossem levantados e documentados por meio da aplicação das dinâmicas de ideação do Design Thinking, promovendo desta forma uma **elicitação de requisitos**. Esta elicitação é materializada pelos documentos gerados pelas dinâmicas de design durante a condução das sessões de Design Thinking, como também pelas histórias de usuário, ambas produzidas pela equipe de design:

*“[...] A gente segue a documentação do nosso cliente. Eles exigem em contrato, fazer as histórias do usuário e registrar e colocar tudo aquilo que foi levantado [...] (p.5)”*.

Neste estudo de caso, também foi observado que o foco principal da aplicação do Design Thinking é justamente o levantamento de requisitos de usuário, mas também são identificados os requisitos de sistema do tipo funcionais. Entretanto, não foi possível ser verificado o levantamento de requisitos de sistema dos tipos não-funcional e de domínio. Uma vez identificados os requisitos pela equipe de design, uma **análise dos requisitos** é feita pela equipe de desenvolvimento para garantir que os produtos concebidos sejam passíveis de implementação através de uma solução de software. Um ponto importante nesta dinâmica é justamente uma certa desconexão das equipes de design com as equipes de desenvolvimento, pois em diversas situações, a solução proposta pela equipe de design não era viável de implementação, sejam por motivos tecnológicos ou por condições de restrições do ambiente do projeto:

*“[...] nem sempre a pessoa que aplica o Design Thinking é uma pessoa envolvida na parte técnica. Às vezes, é aquela pessoa focada na parte de design. E a gente percebeu que isso acaba dificultando um pouco na hora de meter a mão na massa, do desenvolvimento [...] (p.8)”*.

Diante destas dificuldades a organização passou a fazer com que todas as **negociações de requisitos** fossem realizadas durante as execuções das sessões de Design Thinking, evitando assim os problemas anteriormente observados, e mantendo um nível maior de integração entre as equipes:

*“[...] Então, agora, com essas experiências que a gente teve no passado a gente já está um nível melhor. E está a todo o tempo alinhando a equipe que desenvolve com quem está realizando os processos [...] (p.8)”*.

Durante as sessões de Design Thinking são executados testes de usabilidade nos protótipos com os usuários do projeto, visando garantir e identificar se os problemas relatados estão representados e tratados na solução de software proposta:

*[...] apresentação de um protótipo, e acaba fazendo um teste de usabilidade, para estar verificando quais pontos de dores do usuário [...] (p.9)”.*

*“[...] quais são as dificuldades que o usuário tem em relação ao produto que ele mesmo está criando [...] (p.9)”.*

A **documentação de requisitos** é realizada num primeiro momento pelos artefatos produzidos nas dinâmicas de observação & síntese e ideação & prototipação do Design Thinking, como matriz de alinhamento, proto-produto (estudo do produto), mapa de *stakeholders*, diagramas de estratégia, mapa de empatia, proto-*personas* (estudo das *personas*), jornadas do usuário, *warrior moments* (momentos chave que fazem o cliente chegar ao produto final), protótipos e a própria equipe de design constrói as estórias do usuário do método ágil, para posterior priorização do backlog de desenvolvimento e delimitação dos produtos mínimos viáveis (MVP) de cada ciclo de desenvolvimento de software.

Observou-se ainda que a documentação das estórias do usuário é uma obrigatoriedade exigida pelo cliente e não necessariamente um produto gerado pelo Design Thinking. Nestas estórias de usuário são documentados todos os campos dos protótipos e todas as funcionalidades, os quais servirão como base para o desenvolvimento de software. A organização não faz nenhuma complementação nestas documentações para iniciar o desenvolvimento:

*“[...] detalhar todos os campos existentes naquela funcionalidade que foi levantada, o que faz parte daquele fluxo. A gente acaba tendo que documentar tudo que foi feito através do Design Thinking. Não tem como só traçar ali os pontos principais, através de quadro macro [...] (p.5)”.*

Foi relatado que no passado, por não existir um processo de validação de requisitos entre as equipes de design e de desenvolvimento, foi necessário alterar toda uma solução proposta para se adequar à realidade e viabilidade do desenvolvimento de software:

*“[...] eles só se preocuparam em sonhar, mas não se preocuparam após o levantamento, em colocar em prática tudo que o foi solicitado. Então quando a gente verificou que não existiam os serviços, e daí a gente teve que adaptar à realidade atual [...] (p.9)”.*

*“O produto praticamente mudou 99% do que foi levantado [...] (p.9)”.*

Atualmente a **validação dos requisitos** ocorre com o usuário do projeto e as equipes de desenvolvimento durante todo o processo de Design Thinking, justamente para evitar que ocorra a identificação de divergências somente no final do processo. As validações ocorrem pela análise dos protótipos produzidos e são utilizadas todas as evidências formais disponibilizadas pelos usuários como: e-mails, documentos ou qualquer outro tipo de informação como base para garantir a formalidade e acuracidade deste processo.

### **P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis:**

Nesta organização a aplicação do Design Thinking ocorre exclusivamente em **projetos que adotam o modelo ágil:**

*"[...] Aqui só utiliza no ágil [...]" (p.6)".*

### **P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis:**

A organização descreve que a aplicação do Design Thinking é utilizada tanto em softwares desktops com em aplicativos móveis, pois as soluções de software atendidas por esta equipe são baseadas nestas duas famílias de software:

*"[...] Nós temos aplicado nas duas vertentes, até porque quando a gente está desenvolvendo aqui no banco tem tanto web quanto mobile [...]" (p.6)".*

O Quadro 9 demonstra o resumo dos resultados obtidos e analisados no estudo de caso da Organização B:














**Quadro 9 – Resultados obtidos no estudo de caso da Organização B, fonte (O autor, 2020)**

<b>P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos</b>			
<b>Elicitação de Requisitos</b>	<b>Análise e Negociação de Requisitos</b>	<b>Documentação de Requisitos</b>	<b>Validação de Requisitos</b>
Ocorre durante as etapas de ideação & prototipação das sessões de Design Thinking através das aplicações de técnicas de design sem o envolvimento da equipe de desenvolvimento.	Foi observado que ocorrem etapas de análise e negociação de requisitos durante as sessões do Design Thinking com o envolvimento do usuário do projeto e da equipe de desenvolvimento de software.	Os requisitos são documentados pelos subprodutos gerados pelas sessões de Design Thinking e subsequentemente documentados em estórias do usuário pela própria equipe de design.	A validação de requisitos ocorre sempre ao final da aplicação das sessões de Design Thinking através da verificação dos documentos gerados e com o envolvimento do usuário do projeto e da equipe de desenvolvimento.
<b>Tipos de requisitos identificados pelo Design Thinking</b>	Requisitos de usuário e requisitos de sistema do tipo funcional. Não foram observadas evidências de que requisitos de sistema dos tipos não-funcional ou de domínio são tratados pelo Design Thinking.		
<b>Técnicas utilizadas pelo Design Thinking</b>	Matriz de alinhamento, proto-produto, mapa de stakeholders, diagramas de estratégia, mapa de empatia, proto-personas, jornadas do usuário, warrior moments, protótipos funcionais e estórias do usuário.		
<b>P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis</b>			
<b>Modelos de desenvolvimento de software que utilizam o Design Thinking</b>	Somente no modelo ágil.		
<b>P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis</b>			
<b>Famílias de software que utilizam o Design Thinking</b>	Softwares desktops e aplicativos móveis.		










#### 4.4.3.1 Análise dos níveis de influência nos pontos de análise e proposições

O Quadro 10 demonstra o nível de influência identificado em cada um dos pontos de análise e as proposições no estudo de caso da Organização B:

**Quadro 10 – Nível de influência entre pontos de análise e proposições na Organização B, fonte (O autor, 2020)**








Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-01</b> <b>Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking</b>	PA-01-1: Qual foi o objetivo da organização em adotar o Design Thinking em projetos de software?			
	PA-01-2: Qual o modelo de Design Thinking adotado pela organização?			
	PA-01-3: O Design Thinking foi aplicado em quais fases do processo de desenvolvimento do projeto de software (Análise, Projeto, Construção, Testes)?			
	PA-01-4: O Design Thinking é considerado exclusivamente como uma metodologia para se identificar os requisitos necessários para uma solução de software na sua organização?			
	PA-01-5: O Design Thinking é considerado um método de resolução de problemas complexos com foco na solução e não necessariamente em levantamento de requisitos de software?			
	PA-01-6: O projeto evoluiu para as fases de codificação e testes logo após a aplicação das sessões de Design Thinking?			
	PA-01-7: O projeto passou por uma fase de complementação de requisitos logo após a aplicação das sessões de Design Thinking, antes de iniciar as fases de codificação e testes?			
	PA-01-8: Qual o modelo de processo de Engenharia de Software que o projeto utiliza: a) Tradicional: <i>Waterfall, V-model, Spiral, UP-Unified Process</i> , outros; b) Ágil: <i>SCRUM, XP-Extreme Programming, ASD-Adaptive Software Development</i> ?			
	PA-01-9: Qual o tipo de produto de software que é desenvolvido com a utilização do Design Thinking (aplicativos móveis, software <i>desktop</i> , outros)?			

**Quadro 10 – Nível de influência entre pontos de análise e proposições na Organização B, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-02</b> Verificar quais tipos de requisitos de software são identificados pelo Design Thinking	PA-02-1: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do usuário (serviços que a aplicação precisa prover, com suas restrições, sob o ponto de vista do usuário)?			
	PA-02-2: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do sistema (descrições detalhadas das funções do sistema, dos serviços e restrições - definição do que precisa ser desenvolvido sob o ponto de vista de arquitetura)?			
	PA-02-3: Requisitos funcionais foram identificados nos resultados do Design Thinking (declarações dos serviços que o sistema precisa prover, como o sistema precisa reagir a ações dos usuários/outros sistemas e como o sistema precisa se comportar em situações particulares)?			
	PA-02-4: Requisitos não-funcionais foram identificados nos resultados do Design Thinking (restrições nos serviços e/ou nas funções oferecidas pelo sistema como: padrões de usabilidade, tempos limite, restrições no processo de desenvolvimento, padrões de arquitetura/construção dentre outros que não necessariamente representam uma funcionalidade visível na aplicação, mas que precisam ser respeitados)?			
	PA-02-5: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos de domínio (requisitos que vem do domínio do aplicativo e que refletem características desse domínio)?			
	PA-02-6: A utilização do Design Thinking no projeto de software permitiu que fossem identificados requisitos não-funcionais?			
<b>PA-03</b> Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos	PA-03-1: Quais foram as técnicas do Design Thinking que foram aplicadas no projeto de software?			
	PA-03-2: Os requisitos do projeto de software foram identificados exclusivamente pela aplicação do Design Thinking?			
	PA-03-3: Os requisitos do projeto de software foram complementados pela Engenharia de Requisitos após a aplicação do Design Thinking?			



**Quadro 10 – Nível de influência entre pontos de análise e proposições na Organização B, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-04</b> Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos:	PA-04-1: Os requisitos identificados pelo Design Thinking foram analisados e negociados com todas as partes interessadas?			
	PA-04-2: Foram realizadas durante o Design Thinking algum tipo de análise de incompletude dos requisitos?			
	PA-04-3: Foram realizadas durante o Design Thinking algum tipo de análise de ambiguidade dos requisitos?			
	PA-04-4: Foram realizadas durante o Design Thinking algum tipo de análise de inconsistência dos requisitos?			
<b>PA-05</b> Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos	PA-05-1: Quais foram os documentos/produtos gerados pelo Design Thinking no projeto de software?			
	PA-05-2: Os requisitos identificados pelo Design Thinking foram documentados por algum processo formal?			
<b>PA-06</b> Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos	PA-06-1: Os requisitos identificados pelo Design Thinking passaram por um processo de validação por todas as partes interessadas?			

## 4.5 Organização C

### 4.5.1 Caracterização da Organização C

A Organização C é uma empresa de infraestrutura de mercado financeiro, de grande porte e com capital aberto, a qual tem como base de negócio a disponibilização de produtos digitais para desenvolver e viabilizar o mercado financeiro e de capitais para seus clientes. Suas atividades são realizadas por meio da administração e disponibilização de sistemas de negociação, compensação, depósito, registro e liquidação de ativos financeiros, sendo considerada uma das maiores empresas do mundo neste setor.

É uma organização com operações em dois estados brasileiros e com presença internacional em outros países, contando com uma equipe de mais de 2.200 colaboradores.

#### **4.5.2 Contextualização da Adoção do Design Thinking com a Engenharia de Requisitos na Organização C**

Esta organização começou a adotar o Design Thinking em 2013, inicialmente na área de redes sociais, com objetivo de compreender melhor o seu público e suas necessidades. O Design Thinking trouxe um enfoque para extrair informações das áreas de negócio e compartilhar estas informações com outros canais da organização.

Inicialmente foram realizados alguns experimentos com a aplicação do Design Thinking em projetos internos, os quais geraram resultados positivos. Nesta época a alta gestão começou a visualizar os problemas da empresa e as necessidades dos clientes e usuários, o que permitiu a geração de soluções mais apropriadas para o tratamento destes problemas e necessidades.

A aplicação do Design Thinking, nas primeiras experiências desta organização, não se restringia ao contexto de problemas complexos e ocorria em diversas situações dos projetos. Com o tempo, e o ganho de maturidade da equipe de design, o foco de aplicação voltou-se para projetos mais complexos nos quais as áreas de negócio precisavam criar soluções inovadoras, entretanto, não sabiam como fazê-lo. Nestes casos as equipes de design entravam com os workshops de concepção do Design Thinking justamente para extrair uma visão clara e inovadora para o problema que estava sendo tratado:

*“[...] eu utilizo em projetos mais complexos, onde eu não tenho uma visão clara do que a área de negócios quer [...] (p.4)”.*

Entretanto, nem todos os projetos de inovação desta organização utilizam o Design Thinking como uma etapa anterior à Engenharia de Requisitos e muitos produtos são desenvolvidos pelo processo tradicional de Engenharia de Software.

Nas situações em que é possível fazer o aprofundamento no entendimento do problema, as equipes de design aplicam a abordagem completa do duplo diamante do Design Thinking composta por observação & síntese e ideação & prototipação, entretanto, quando isso não é possível por imposição do tempo disponível para o projeto, são aplicadas partes do processo de observação & síntese e ideação & prototipação para garantir que as equipes consigam desenvolver suas ideias:

*“[...] Tem projeto que eu não consigo fazer de ponto a ponto, o diamante duplo. Às vezes eu pego pedaços dele, e aplico no projeto [...] (p.8)”*

As equipes de design fazem um elo de ligação entre as áreas de negócio e as áreas de tecnologia, trazendo um enfoque ligado à inovação e experiência do usuário (UX) para os produtos desenvolvidos.

O modelo de Design Thinking adotado na organização é baseado em várias escolas, incluindo a de Tim Brown (BROWN, 2008). A organização desenvolveu um kit próprio de ferramentas, baseados em diversas escolas e experiências de seu time de design, onde as ferramentas mais adequadas são escolhidas com base nas necessidades do negócio e no perfil do público alvo. Mesmo com a criação deste kit de ferramentas, essencialmente a abordagem é sempre baseada na exploração do espaço do problema e na exploração do espaço da solução do Design Thinking, também conhecida como duplo diamante. A área de design relata que em algumas áreas, o público alvo não é informado de que está participando de sessões de Design Thinking, pois percebe-se um certo tipo de resistência na adoção de técnicas de cocriação.

#### **4.5.3 Resultados das Análises das Proposições na Organização**

##### **P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos:**

A organização utiliza o Design Thinking como ferramenta para a concepção de produtos inovadores, a qual visa extrair dos usuários e clientes os principais insumos e diretrizes (requisitos de negócio) que caracterizarão estes produtos. Durante os workshops de Design Thinking, os insumos e diretrizes derivam em requisitos de usuário e protótipos, os quais vão sendo agrupados em mínimos produtos viáveis (MVP) para o posterior detalhamento de requisitos de sistema e desenvolvimento da solução. A decisão de não tratar requisitos de sistema dentro das dinâmicas do Design Thinking é justamente para não limitar a inovação nos produtos que estão sendo concebidos.

Uma vez que os requisitos de usuário são identificados pelo Design Thinking, a **elicitação de requisitos** de sistema, sejam funcionais ou não-funcionais, é conduzida por um analista de requisitos. Isso sempre ocorre após o término dos workshops de Design Thinking, tendo por objetivo a identificação destes requisitos e

o refinamento do mínimo produto viável (MVP) de cada uma das etapas de desenvolvimento de software que se sucederão:

*“[...] Design Thinking a gente só vai fazer concepção e sai com uma ideia do produto: eu quero isso. Fechou. Aí depois, sim, a gente envolve uma analista de requisito, que faz o match entre o produto e tecnologia, e desenvolvimento [...] (p.5)”*

*“[...] A partir do MVP, sim, aí a gente cruza com requisito, onde a gente vai fazer o requisito das funcionalidades, e aí fechar o MVP para entrar em desenvolvimento [...] (p.5)”*

Esta dinâmica de elicitação de requisitos de usuário sendo conduzida pelo Design Thinking e de requisitos de sistema sendo conduzida pelo processo de Engenharia de Requisitos é a condição mais comum no processo de desenvolvimento de softwares inovadores nesta organização. Entretanto, existem casos em que não é possível se realizar uma etapa de Engenharia de Requisitos após o término dos workshops de Design Thinking. Nestes casos, as equipes de desenvolvimento de software já iniciam a construção do software tomando como base apenas os requisitos de negócio e os protótipos gerados pelo Design Thinking:

*“Tem projetos em que não dá tempo de a gente fazer tudo que a gente gostaria, a gente já tem que partir para a produção. O que a gente fez aqui, nesses casos: a gente fez um design system que é a componentização de todos os elementos de um sistema, ou da interface web. Então ali eu já tenho uma biblioteca, onde os desenvolvedores já conseguem desenvolver, sem o requisito, eles já têm muitos insumos ali. Mas às vezes fica faltando a parte de requisito [...] (p.6)”*

Alguns tipos de requisitos de sistema do tipo não-funcional são previamente identificados pelo Design Thinking, como requisitos de segurança e tecnologia, mas principalmente requisitos ligados às questões de usabilidade.

Durante as dinâmicas de Design Thinking e algumas vezes em campo após as sessões, existe um processo de **análise e negociação** dos requisitos identificados pelo processo de ideação & prototipação do Design Thinking. Isso garante que não tenha ocorrido algum tipo de omissão de requisitos importantes para a concepção do produto:

*“[...] Durante o teste, a gente consegue, aí sim, enxergar o nosso produto pelo olhar do usuário, e aí aparecem coisas que às vezes o time não captou [...] (p.14)”*

A **documentação de requisitos** de negócio se dá pela elaboração de diversos artefatos resultantes da aplicação das técnicas de Design Thinking como por exemplo: canvas de empatia, personas, jornada de usuário (service blueprint), sketching (o qual

é um dos mais utilizados), relatório de dores e oportunidades, definições de mínimo produto viável (MVP), protótipos e layouts de telas. Tudo isso concebido dentro de uma plataforma de design system desenvolvida pela própria área de design, com o intuito de facilitar o posterior desenvolvimento do software. Já a documentação de requisitos de sistema segue os artefatos exigidos pelo processo Engenharia de Requisitos da organização.

Ao final dos workshops de Design Thinking os documentos, protótipos e layouts de telas desenvolvidos são apresentados novamente aos participantes das dinâmicas, com o intuito de realizar a **validação final de requisitos** do produto, conforme observado em:

*"[...] a gente volta, até mostrando para o time que participou do processo de cocriação, para falar assim: "esse aqui é o produto que vai entrar na rua. Está de acordo com o que foi criado?". A gente dá uma satisfação para o pessoal que participou, sim. Até para validar [...] (p.16)".*

## **P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis:**

Foi relatado que o processo de desenvolvimento de software desta organização **segue o modelo waterfall** e que todos os projetos que aplicaram o Design Thinking seguiam este modelo. Existe um plano corporativo de migração do modelo waterfall para o método ágil, entretanto esta transição de modelos ainda está sendo implantada na organização:

*"[...] A empresa, é waterfall. Mas está com transformação ágil [...] (p.6)".*

## **P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis:**

As aplicações desenvolvidas por esta organização, com o apoio do Design Thinking, são predominantemente aplicações desktops, sendo que a utilização em aplicações móveis é menos frequente. Isso ocorre não por uma limitação da aplicabilidade desta metodologia nestas famílias de software, mas sim pela característica dos sistemas desta organização, os quais tratam de diversas informações ao mesmo tempo. Desta forma o consumo destas informações em sistemas móveis torna-se inviável:

*"[...] o que a gente menos usa é mobile. É mais sistema desktop. [...] (p.10)".*

O Quadro 11 demonstra o resumo dos resultados obtidos e analisados no estudo de caso da Organização C:














**Quadro 11 – Resultados obtidos no estudo de caso da Organização C, fonte (O autor, 2020)**

<b>P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos</b>			
<b>Elicitação de Requisitos</b>	<b>Análise e Negociação de Requisitos</b>	<b>Documentação de Requisitos</b>	<b>Validação de Requisitos</b>
O Design Thinking auxilia a elicitação de requisitos de usuário. A derivação dos requisitos de negócios em requisitos de sistema é conduzida fora das dinâmicas de design por um analista de requisitos.	Foi observado que ocorrem etapas de análise e negociação de requisitos, mas somente para requisitos de usuário durante as sessões de Design Thinking, e em algumas vezes em campo após as sessões.	Os requisitos de usuário são documentados nos produtos gerados pelas sessões de Design Thinking.	A validação de requisitos de sistema ocorre sempre ao final da aplicação das sessões de Design Thinking para a aprovação do produto concebido pela equipe participante do workshop.
<b>Tipos de requisitos identificados pelo Design Thinking</b>		Requisitos de usuário e eventualmente alguns requisitos de sistema do tipo não-funcional como por exemplo: usabilidade, segurança e tecnologia.	
<b>Técnicas utilizadas pelo Design Thinking</b>		Canvas de empatia, personas, jornada de usuário (service blueprint), sketching, relatório de dores e oportunidades, definições de mínimo produto viável (MVP), protótipos e layouts.	
<b>P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis</b>			
<b>Modelos de desenvolvimento de software que utilizam o Design Thinking</b>		Somente no modelo waterfall.	
<b>P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis</b>			
<b>Famílias de software que utilizam o Design Thinking</b>		Preponderantemente em softwares desktops e poucos casos em aplicativos móveis.	










#### 4.5.3.1 Análise dos níveis de influência nos pontos de análise e proposições

O Quadro 12 demonstra o nível de influência identificado em cada um dos pontos de análise e as proposições no estudo de caso da Organização C:

**Quadro 12 – Nível de influência entre pontos de análise e proposições na Organização C, fonte (O autor, 2020)**








Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-01</b> <b>Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking</b>	PA-01-1: Qual foi o objetivo da organização em adotar o Design Thinking em projetos de software?			
	PA-01-2: Qual o modelo de Design Thinking adotado pela organização?			
	PA-01-3: O Design Thinking foi aplicado em quais fases do processo de desenvolvimento do projeto de software (Análise, Projeto, Construção, Testes)?			
	PA-01-4: O Design Thinking é considerado exclusivamente como uma metodologia para se identificar os requisitos necessários para uma solução de software na sua organização?			
	PA-01-5: O Design Thinking é considerado um método de resolução de problemas complexos com foco na solução e não necessariamente em levantamento de requisitos de software?			
	PA-01-6: O projeto evoluiu para as fases de codificação e testes logo após a aplicação das sessões de Design Thinking?			
	PA-01-7: O projeto passou por uma fase de complementação de requisitos logo após a aplicação das sessões de Design Thinking, antes de iniciar as fases de codificação e testes?			
	PA-01-8: Qual o modelo de processo de Engenharia de Software que o projeto utiliza: a) Tradicional: <i>Waterfall, V-model, Spiral, UP-Unified Process</i> , outros; b) Ágil: <i>SCRUM, XP-Extreme Programming, ASD-Adaptive Software Development</i> ?			
	PA-01-9: Qual o tipo de produto de software que é desenvolvido com a utilização do Design Thinking (aplicativos móveis, software <i>desktop</i> , outros)?			

**Quadro 12 – Nível de influência entre pontos de análise e proposições na Organização C, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-02</b> Verificar quais tipos de requisitos de software são identificados pelo Design Thinking	PA-02-1: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do usuário (serviços que a aplicação precisa prover, com suas restrições, sob o ponto de vista do usuário)?			
	PA-02-2: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos do sistema (descrições detalhadas das funções do sistema, dos serviços e restrições - definição do que precisa ser desenvolvido sob o ponto de vista de arquitetura)?			
	PA-02-3: Requisitos funcionais foram identificados nos resultados do Design Thinking (declarações dos serviços que o sistema precisa prover, como o sistema precisa reagir a ações dos usuários/outros sistemas e como o sistema precisa se comportar em situações particulares)?			
	PA-02-4: Requisitos não-funcionais foram identificados nos resultados do Design Thinking (restrições nos serviços e/ou nas funções oferecidas pelo sistema como: padrões de usabilidade, tempos limite, restrições no processo de desenvolvimento, padrões de arquitetura/construção dentre outros que não necessariamente representam uma funcionalidade visível na aplicação, mas que precisam ser respeitados)?			
	PA-02-5: A utilização do Design Thinking no projeto de software permitiu que fossem identificados os requisitos de domínio (requisitos que vem do domínio do aplicativo e que refletem características desse domínio)?			
	PA-02-6: A utilização do Design Thinking no projeto de software permitiu que fossem identificados requisitos não-funcionais ?			
<b>PA-03</b> Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos	PA-03-1: Quais foram as técnicas do Design Thinking que foram aplicadas no projeto de software?			
	PA-03-2: Os requisitos do projeto de software foram identificados exclusivamente pela aplicação do Design Thinking?			
	PA-03-3: Os requisitos do projeto de software foram complementados pela Engenharia de Requisitos após a aplicação do Design Thinking?			



**Quadro 12 – Nível de influência entre pontos de análise e proposições na Organização C, fonte (O autor, 2020)**

Proposições →		P1 O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	P2 O Design Thinking é utilizado com projetos que adotam os métodos ágeis	P3 O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis
Pontos de análise ↓				
<b>PA-04</b> Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos:	PA-04-1: Os requisitos identificados pelo Design Thinking foram analisados e negociados com todas as partes interessadas?			
	PA-04-2: Foram realizadas durante o Design Thinking algum tipo de análise de incompletude dos requisitos?			
	PA-04-3: Foram realizadas durante o Design Thinking algum tipo de análise de ambiguidade dos requisitos?			
	PA-04-4: Foram realizadas durante o Design Thinking algum tipo de análise de inconsistência dos requisitos?			
<b>PA-05</b> Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos	PA-05-1: Quais foram os documentos/produtos gerados pelo Design Thinking no projeto de software?			
	PA-05-2: Os requisitos identificados pelo Design Thinking foram documentados por algum processo formal?			
<b>PA-06</b> Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos	PA-06-1: Os requisitos identificados pelo Design Thinking passaram por um processo de validação por todas as partes interessadas?			

#### 4.6 Resumo dos Resultados Obtidos nos Estudos de Caso

O Quadro 13 demonstra o resumo dos resultados obtidos e analisados nos estudos de caso desta pesquisa:

**Quadro 13 – Resultados obtidos nos estudos de caso desta pesquisa, fonte (O autor, 2020)**

Ponto de observação	Organização		
	A	B	C
<b>P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos</b>			
Elicitação de Requisitos	😊	😊	😊
Requisitos de Usuário	😊	😊	😊
Requisitos de Sistema:			
Requisitos Funcionais	😊	😊	😞
Requisitos Não-Funcionais	😊	😞	😊 <small>(alguns tipos de requisitos como: usabilidade, segurança e tecnologia)</small>
Requisitos de Domínio	😞	😞	😞
Análise e Negociação de Requisitos	😊	😊	😞 <small>(somente para requisitos de usuário)</small>
Documentação de Requisitos	😊	😊	😞 <small>(somente para requisitos de usuário)</small>
Validação de Requisitos	😊	😊	😞 <small>(somente para requisitos de usuário)</small>
Técnicas utilizadas pelo Design Thinking	Documentos de especificação de produtos, personas, jornadas do usuário e protótipos.	Matriz de alinhamento, proto-produto, mapa de stakeholders, diagramas de estratégia, mapa de empatia, proto-personas, jornadas do usuário, warrior moments, protótipos funcionais e estórias do usuário.	Canvas de empatia, personas, jornada de usuário (service blueprint), sketching, relatório de dores e oportunidades, definições de mínimo produto viável (MVP), protótipos e layouts.
<b>P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis</b>			
Método Ágil	😊	😊	😞
Modelo Waterfall	😊	😞	😊
<b>P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis</b>			
Software desktop	😊	😊	😊
Aplicativos móveis	😊	😊	😞 <small>(poucos casos)</small>
<b>Legenda:</b>			
😊 A aplicação do Design Thinking foi <b>amplamente</b> identificada na organização para este ponto de observação			
😞 A aplicação do Design Thinking foi identificada, mas de forma <b>parcial ou incompleta</b> na organização para este ponto de observação			
😞 A aplicação do Design Thinking <b>não foi identificada</b> na organização para este ponto de observação			

## 4.7 Considerações sobre o capítulo

Neste capítulo foram apresentados os resultados obtidos pelos estudos de caso realizados em três organizações. Os dados destes estudos de caso foram analisados individualmente, tomando como base as premissas e a correlação das etapas da Engenharia de Requisitos com as sessões de Design Thinking. Esta análise foi conduzida desta forma com o intuito de promover um entendimento de como estas duas abordagens estão se relacionando nas organizações analisadas.

## CAPÍTULO 5 - DISCUSSÃO DOS RESULTADOS

O objetivo deste capítulo é apresentar uma discussão dos resultados identificados nos estudos de casos conduzidos por esta pesquisa, correlacionando-os ao referencial teórico, às proposições e aos respectivos conceitos de apoio. O resultado desta discussão é demonstrar como o Design Thinking auxilia a Engenharia de Requisitos em projetos de software, dentro das organizações analisadas.

### 5.1 Análise das Proposições

O Quadro 14 apresenta um resumo do nível de identificação dos pontos de análise nas organizações analisadas, bem como sua respectiva ligação às proposições desta pesquisa. Ele servirá como base para a análise e discussão individualizada, de cada um dos pontos de análise, que serão apresentadas nas próximas sessões deste capítulo:

**Quadro 14 – Síntese da verificação dos pontos de análise nas organizações, fonte (O autor, 2020)**

Pontos de análise	Proposições			Organização		
	P1	P2	P3	A	B	C
PA-01 Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking	◆	◆	◆	😊	😊	😊
PA-02 Verificar quais tipos de requisitos de software são identificados pelo Design Thinking	◆			😊	😊	😊
PA-03 Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos	◆			😊	😊	😊
PA-04 Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos	◆			😊	😊	😊
PA-05 Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos	◆			😊	😊	😊
PA-06 Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos	◆			😊	😊	😊
<b>Legenda:</b>						
◆ Ponto de análise <b>correlacionado</b> à proposição						
😊 Ponto de análise foi <b>amplamente</b> identificado na organização						
😐 Ponto de análise foi identificado, mas de forma <b>parcial</b> na organização						
☹️ Ponto de análise <b>não foi identificado</b> na organização						

Já o Quadro 15 apresenta o nível de verificação das proposições em cada uma das organizações, e também servirá como base para a análise e discussão individualizada de cada uma das proposições, que serão apresentadas nas próximas sessões deste capítulo:

**Quadro 15 – Síntese da verificação das proposições nas organizações, fonte (O autor, 2020)**

Proposições	Organização		
	A	B	C
P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos	😊	😊	😊
P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis	😊	😊	😞
P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis	😞	😊	😞
<b>Legenda:</b>			
😊 Proposição foi <b>amplamente</b> validada na organização			
😞 Proposição foi validada, mas de forma <b>parcial</b> na organização			
😞 Proposição <b>não foi validada</b> na organização			

### 5.1.1 Proposição 1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos

O Quadro 16 descreve todos os conceitos de apoio utilizados para a definição da proposição P1 e a verificação ou não da identificação dos pontos de análise desta proposição no estudo de caso de cada uma das organizações investigadas:

**Quadro 16 – Síntese da proposição P1 nas organizações, fonte (O autor, 2020)**

P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos
<p>Conceitos de apoio à proposição:</p> <ul style="list-style-type: none"> <li>• Design Thinking provê um processo estruturado para a elicitação de requisitos para problemas complexos (HEHN; UEBERNICKEL, 2018).</li> <li>• Se você precisa compreender em detalhe os contextos de negócio e dos usuários, utilize o Design Thinking (HEHN et al., 2020).</li> <li>• Design Thinking é consistente com as práticas iniciais de elicitação da Engenharia de Requisitos (VETTERLI et al., 2013).</li> <li>• Design Thinking emprega a sensibilidade e as técnicas dos designers para corresponder às necessidades dos clientes dentro daquilo que é tecnologicamente viável (VALENTIM et al., 2018).</li> <li>• Design Thinking é consistente com as primeiras práticas das atividades de elicitação de requisitos, como por exemplo, rápida prototipação e envolvimento dos interessados, como também observado pelos métodos ágeis de desenvolvimento (SOUZA; SILVA, 2015).</li> <li>• Nós vislumbramos o Design Thinking como uma “arma externa” para a compreensão de problemas complexos na Engenharia de Requisitos, enquanto a Engenharia de Requisitos oferece um framework forte e integrado para o Design Thinking dentro do ciclo de desenvolvimento de software (HEHN; UEBERNICKEL, 2018).</li> <li>• Nos últimos anos, diversos autores têm sido enfáticos na importância de se vislumbrar a engenharia de requisitos como um processo criativo de resolução de problemas (SOUZA; SILVA, 2015).</li> <li>• Design Thinking complementa as técnicas tradicionais da engenharia de requisitos de sistemas como uma análise de requisitos dirigida por protótipos. Design Thinking vai além da elicitação de requisitos nos impedindo de ser restringido por preconceitos de soluções de software de forma isolada (CARROLL; RICHARDSON, 2016).</li> <li>• (SOLEDAD et al., 2013) apresentam uma experiência em refinar requisitos através do uso do Design Thinking para um sistema de gestão do conhecimento (SOUZA; SILVA, 2015).</li> <li>• Design Thinking ajuda na compreensão das necessidades reais iniciais, por exemplo tornar o problema endereçável, enquanto a Engenharia de Requisitos provê métodos para projetar a arquitetura, definir requisitos de segurança, e assim por diante (HEHN; UEBERNICKEL, 2018).</li> </ul>

**Quadro 16 – Síntese da proposição P1 nas organizações, fonte (O autor, 2020)**

<b>P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos</b>			
<ul style="list-style-type: none"> <li>Design Thinking provê uma metodologia para se elicitar as necessidades dos clientes e não requisitos, produzindo uma sequência rápida e simples de protótipos que eventualmente convergem para uma solução inovadora (VETTERLI et al., 2013).</li> </ul>			
<b>Pontos de análise</b>	<b>Organização</b>		
	<b>A</b>	<b>B</b>	<b>C</b>
PA-01 Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking	😊	😊	😊
PA-02 Verificar quais tipos de requisitos de software são identificados pelo Design Thinking	😊	😊	😊
PA-03 Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos	😊	😊	😊
PA-04 Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos	😊	😊	😊
PA-05 Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos	😊	😊	😊
PA-06 Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos	😊	😊	😊
<b>Legenda:</b>			
😊 Ponto de análise foi <b>amplamente</b> identificado na organização			
😐 Ponto de análise foi identificado, mas de forma <b>parcial ou incompleta</b> na organização			
😞 Ponto de análise <b>não foi identificado</b> na organização			

### **PONTO DE ANÁLISE: PA-01 - Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking**

A aplicação do Design Thinking em problemas complexos possibilita transformá-los em problemas mais bem definidos e compreensíveis, e a partir daí entrar com as práticas bem estabelecidas da Engenharia de Requisitos para seguir com o desenvolvimento do software (HEHN; UEBERNICKEL, 2018). Neste sentido, o ponto de análise “PA-01 Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking” visa compreender, nas organizações analisadas, os principais fatores que influenciaram a adoção do Design Thinking com o desenvolvimento de software, especificamente junto com a etapa de Engenharia de Requisitos.

Com base nos resultados do ponto de análise PA-01 foram identificados os seguintes fatores:

- Agregar mais valor na relação com os clientes;
- Impulsionar a inovação pelo desenvolvimento de software;
- Aumentar a velocidade de criação de novos produtos baseados em soluções de software;
- Melhorar a comunicação entre áreas de negócio e de tecnologia;
- Identificar as necessidades dos usuários através de um aprofundamento dos problemas, complexos ou não, que estão sendo enfrentados;
- Melhorar alinhamento do que é tecnologicamente viável; e
- Melhorar o processo de levantamento de requisitos.

O fator para impulsionamento da inovação no desenvolvimento de software pela utilização do Design Thinking, onde tem-se uma engenharia de requisitos como um processo criativo de resolução de problemas, sem conceitos pré-estabelecidos, e com a possibilidade de encontrar soluções inovadoras já foi observado nos estudos de (VETTERLI et al., 2013; SOUZA; SILVA, 2015; CARROLL; RICHARDSON, 2016). Por outro lado, o aumento na velocidade de criação de novos produtos baseados em soluções de software está presente no estudo de (VETTERLI et al., 2013) dentro de um contexto de aplicações móveis. O aprimoramento da comunicação entre as áreas com o uso do Design Thinking, permite atacar um dos principais problemas na Engenharia de Requisitos descritos por (JALIL et al., 2019). Já a identificação das necessidades dos usuários, através de um aprofundamento dos problemas, está diretamente relacionada ao conceito de se ter o Design Thinking como uma “arma externa” para a compreensão destes tipos de problemas enfrentados pela Engenharia de Requisitos (HEHN; UEBERNICKEL, 2018). Para melhorar o alinhamento do que é tecnologicamente viável é necessário trazer uma visão, pelo Design Thinking, daquilo que corresponda às necessidades dos clientes, mas dentro do que é tecnologicamente possível, conforme proposto pelo estudo de (VALENTIM et al., 2018). Por fim, a melhoria no processo de levantamento de requisitos considerando a utilização do Design Thinking como uma ferramenta auxiliar já foi amplamente apresentado pelos estudos de (VETTERLI et al. 2013; SOUZA; SILVA, 2015; CARROLL; RICHARDSON, 2016; HEHN; UEBERNICKEL, 2018).

Pela análise dos resultados do ponto de análise PA-01, foi possível verificar que apenas o fator “Agregar mais valor na relação com os clientes” ainda não havia sido citado na literatura. Todos os outros fatores já foram explorados em estudos anteriores e pode-se concluir que os mesmos fatores que estão motivando organizações a utilizarem o Design Thinking com a Engenharia de Requisitos, foram verificados nas organizações dos estudos de caso desta pesquisa.

### **PONTO DE ANÁLISE: PA-02 - Verificar quais tipos de requisitos de software são identificados pelo Design Thinking**

A proposta de que o Design Thinking tem apoiado a identificação de requisitos, com uma grande afinidade para requisitos de usuário e/ou necessidades do negócio (VETTERLI et al., 2013; SOUZA; SILVA, 2015; HEHN; UEBERNICKEL, 2018;

VALENTIM et al., 2018; HEHN et al., 2020) e a proposta do uso do Design Thinking para o refinamento de requisitos (SOLEDA et al., 2013), são os conceitos de apoio utilizados para a definição do ponto de análise “PA-02 Verificar quais tipos de requisitos de software são identificados pelo Design Thinking”. Este ponto de análise tem por objetivo compreender quais tipos de requisitos, baseados na taxonomia da Engenharia de Software, estão sendo identificados pelo Design Thinking.

O Quadro 17 demonstra uma síntese dos tipos de requisitos identificados pelo Design Thinking, observados durante os estudos de caso desta pesquisa:

**Quadro 17 – Tipos de requisitos levantados pelo Design Thinking, fonte (O autor, 2020)**

Tipo de Requisito	Organização		
	A	B	C
Requisitos de Usuário	😊	😊	😊
Requisitos de Sistema			
Requisitos Funcionais	😊	😊	😞
Requisitos Não-Funcionais	😊	😞	😊 (alguns tipos de requisitos como: usabilidade, segurança e tecnologia)
Requisitos de Domínio	😞	😞	😞
<b>Legenda:</b>			
😊 O tipo de requisito foi <b>amplamente</b> identificado pelo Design Thinking na organização			
😊 O tipo de requisito foi identificado, mas de forma <b>parcial ou incompleta</b> pelo Design Thinking na organização			
😞 O tipo de requisito <b>não foi identificado</b> pelo Design Thinking na organização			

O tratamento de requisitos de usuário foi observado em todas as organizações, o que sugere a efetividade da aplicação do Design Thinking no entendimento das necessidades e problemas do negócio, conforme já sugerido por (LINDBERG et al., 2011; CARROLL; RICHARDSON, 2016; SOUZA et al., 2017; VALENTIM et al., 2018; HEHN et al., 2020).

A identificação de requisitos de sistema pelo Design Thinking foi apresentada no estudo de (VETTERLI et al., 2013) o qual descreve o Design Thinking como um guia para se chegar ao espaço de solução de maneira mais efetiva. Já (SOUZA; SILVA, 2015) e (HEHN; UEBERNICKEL, 2018) utilizaram o Design Thinking para a elicitación dos requisitos de sistema de uma aplicação de software. Para os requisitos de sistema, nesta pesquisa, apenas duas Organizações fazem algum tipo de tratamento. A Organização A é a que tem uma maior abrangência nos tipos de requisitos tratados, possivelmente pelo fato de que as equipes de desenvolvimento atuam junto com as equipes de design durante as sessões de Design Thinking. Na Organização B os requisitos de sistema funcionais são identificados pela própria equipe de design, mas os requisitos de sistema não-funcionais e de domínio não são

tratados durante as sessões de Design Thinking. Já na Organização C o Design Thinking se restringe à identificação de requisitos de usuário e não são identificados requisitos de sistema funcionais e de domínio, sendo identificado apenas alguns tipos de requisitos de sistema não-funcionais. Um ponto comum na identificação de requisitos não-funcionais nas organizações, é que esta identificação tem um foco muito mais voltado para aspectos de usabilidade, e não uma análise mais abrangente de todas as categorias de requisitos não-funcionais.

Pela análise dos resultados do ponto de análise PA-02, foi possível verificar que em nenhuma das organizações são identificados todos os tipos de requisitos da Engenharia de Requisitos pelas sessões de Design Thinking, mas que em todas as organizações estão sendo identificados requisitos de usuário. Pode-se concluir que, para estas organizações, o Design Thinking tem auxiliado a Engenharia de Requisitos na identificação de requisitos, mesmo pelo fato de que nem todas as categorias de requisitos estejam sendo descobertas pelas sessões de Design Thinking.

### **PONTO DE ANÁLISE: PA-03 - Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos**

A ideia de que o Design Thinking contribui para a elicitação de requisitos já foi discutida em vários estudos. Nos trabalhos de (VETTERLI et al., 2013; SOUZA; SILVA, 2015; HEHN et al., 2020) são descritos a relação direta do Design Thinking com as primeiras atividades de descobertas das necessidades dos usuários através de um processo criativo, enquanto (HEHN; UEBERNICKEL, 2018) demonstra a aplicação desta abordagem, como uma técnica complementar à Engenharia de Requisitos, na descoberta de requisitos para problemas complexos. Para (VALENTIM et al., 2018) o Design Thinking traz a sensibilidade e as técnicas dos designers para uma solução a ser desenvolvida, mas equilibrando com o que é tecnologicamente viável. Estas evidências fundamentaram o ponto de análise “PA-03 Compreender a aplicação do Design Thinking na etapa de Elicitação de Requisitos” cujo objetivo é validar a existência das condições de elicitação de requisitos nos estudos de caso desta pesquisa.

Foi observado que, tanto na Organização A quanto na Organização B, o Design Thinking está sendo aplicado para se ter uma maior assertividade nas necessidades dos usuários, condição esta já relatada por (VETTERLI et al., 2013; SOUZA; SILVA, 2015). Assim como relatado no estudo de (VALENTIM et al., 2018), na Organização



A as equipes de desenvolvimento atuam concomitantemente com as equipes de design durante a elicitação de requisitos, para se ter uma clareza de escopo e para garantir que as soluções propostas pelas equipes de design são tecnologicamente viáveis. Um ponto importante observado na Organização B é que todos os requisitos necessários para o desenvolvimento de software são gerados pelas próprias equipes de design o que sugere, nesta organização, uma completa substituição da etapa de elicitação de requisitos pelas sessões de Design Thinking. Para a Organização C o Design Thinking é aplicado para extrair os insumos e diretrizes para a criação de novos produtos, sendo restrito à elicitação de requisitos de negócio, justamente para não limitar a inovação nas soluções idealizadas. Isso permite à organização que o desenho das soluções propostas não seja influenciado pelas restrições impostas pelos preconceitos das soluções de software já existentes, conforme também observado pelo estudo de (CARROLL; RICHARDSON, 2016).

Com base nos resultados observados nas organizações por meio do ponto de análise PA-03, é possível afirmar que o Design Thinking tem contribuindo de maneira bastante efetiva na elicitação de requisitos, principalmente para requisitos de negócio, para garantir uma maior assertividade nas soluções de software.

#### **PONTO DE ANÁLISE: PA-04 - Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos**

No que se refere à análise e negociação de requisitos, etapa subsequente à etapa de elicitação, o conceito de condução de uma análise de requisitos à partir de protótipos gerados (CARROLL; RICHARDSON, 2016) bem como o refinamento de requisitos através do Design Thinking (SOLEDADE et al., 2013) estão diretamente ligados à referência de que o Design Thinking poderia ser considerado como uma moderna forma de Engenharia de Requisitos (VETTERLI et al., 2013; HEHN; UEBERNICKEL, 2018). Com base nestes conceitos de apoio foi formulado o ponto de análise “PA-04 Compreender a aplicação do Design Thinking na etapa de Análise e Negociação de Requisitos”, cujo objetivo é investigar se as atividades de análise e negociação de requisitos estão sendo conduzidas pelo Design Thinking.

Nas Organizações A e B foi identificado que ocorre algum tipo de análise e negociação de requisitos, sempre durante a fase de ideação & prototipação do Design Thinking, mas com um foco voltado à viabilidade tecnológica da solução idealizada. Nestas organizações este tipo de análise é sempre conduzido pelas equipes de design

e equipes de desenvolvimento de software, com base na viabilidade e nas restrições tecnológicas do projeto. A apresentação dos protótipos gerados pelo Design Thinking para a condução de uma análise com o usuário (CARROLL; RICHARDSON, 2016) foi observada nas três organizações, cujo foco foi a verificação da adequação da solução proposta com o problema endereçado.

Desta forma, pode-se concluir por meio das análises dos resultados do ponto de análise PA-04, que nestas organizações algum tipo de análise e negociação de requisitos está sendo realizada pelo Design Thinking.

### **PONTO DE ANÁLISE: PA-05 - Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos**

A proposta de que uma documentação gerada pelo Design Thinking é consistente com a Engenharia de Requisitos (VERTELLI et al., 2013), bem como o conceito de que o Design Thinking vai além da elicitação de requisitos (CARROLL; RICHARDSON, 2016), formam os conceitos de apoio utilizados para a formulação do ponto de análise “PA-05 Compreender a aplicação do Design Thinking na etapa de Documentação de Requisitos”, cujo objetivo é identificar o nível de documentação de requisitos apresentado pelas sessões de Design Thinking.

Em todas as organizações foi observado que o principal produto gerado pelas sessões de Design Thinking são os protótipos, os quais sempre são utilizados como base para as fases subsequentes do desenvolvimento de software. Para (HEHN et al., 2020) a documentação de requisitos pode ocorrer pelas equipes de desenvolvimento durante ou posterior às sessões de Design Thinking, e foi observado que na Organização A a documentação de requisitos ocorre concomitantemente com as sessões de Design Thinking, uma vez que as equipes de desenvolvimento acabam participando deste processo e auxiliando as equipes de design na produção dos documentos. Nesta organização somente são gerados documentos complementares, nos padrões da Engenharia de Requisitos, quando existe uma exigência contratual do cliente. Uma vez encerrada as sessões de Design Thinking as equipes de desenvolvimento já podem iniciar o processo de construção de software tomando como base somente os artefatos produzidos por estas sessões. Para a Organização B a equipe de design gera toda a documentação de requisitos com base nos artefatos produzidos pelas técnicas do Design Thinking. Na sequência esta mesma equipe também gera histórias do usuário baseadas no padrão dos métodos ágeis, como já

descrito por (VERTELLI et al., 2013), o que muitas vezes acaba gerando inconsistências para o desenvolvimento do software pois não são equipes especializadas em engenharia de software. Já na Organização C a documentação de requisitos é feita pela equipe de design e é materializada somente para os requisitos do usuário, por meio de artefatos produzidos pelas sessões de Design Thinking. Nesta organização, após o encerramento das sessões de Design Thinking, uma equipe de analistas de requisitos faz toda a documentação de requisitos de sistema, nos padrões da Engenharia de Requisitos, para o início das etapas de construção de software. Esta abordagem utilizada pela Organização C é muito similar à proposta apresentada por (HEHN et al., 2020) na qual o Design Thinking é utilizado para primeiramente realizar um melhor entendimento do problema, para que posteriormente a Engenharia de Requisitos promova a documentação dos requisitos da solução.

O Quadro 18 apresenta os principais documentos são produzidos pelo Design Thinking nestas organizações:

**Quadro 18 – Tipos de documentos gerados pelo Design Thinking, fonte (O autor, 2020)**

Documentos	Organização		
	A	B	C
Canvas de empatia	☒	☒	☺
Definições de MVP	☒	☺	☺
Diagrama de estratégia	☒	☺	☒
Especificação de produto	☺	☒	☒
Estórias do usuário	☒	☺	☒
Jornadas do usuário (service blueprint)	☺	☺	☺
Layout de telas	☒	☒	☺
Mapa de empatia	☒	☺	☒
Mapa de stakeholders	☒	☺	☒
Matriz de alinhamento	☒	☺	☒
Personas	☺	☒	☺
Proto-persona (estudo das personas)	☒	☺	☒
Proto-produto (estudo do produto)	☒	☺	☒
Protótipo funcional	☺	☺	☺
Sketching	☒	☒	☺
Warrior moments/Relatório de dores	☒	☺	☺
<b>Legenda:</b>			
☺ O tipo de artefato foi <b>produzido</b> pelo Design Thinking na organização			
☒ O tipo de artefato <b>não foi produzido</b> pelo Design Thinking na organização			

Já o Quadro 19 apresenta um resumo se há a necessidade, ou não, de complementação da documentação de requisitos pela Engenharia de Requisitos após as sessões de Design Thinking para permitir o início do desenvolvimento do software:

**Quadro 19 – Abrangência da documentação de requisitos pelo Design Thinking, fonte (O autor, 2020)**

<b>Orga- nização</b>	<b>Documentação de requisitos de negócio</b>	<b>Documentação requisitos de sistema</b>	<b>Quando inicia o desenvolvimento de software</b>
<b>A</b>	Materializados nos documentos gerados pelo Design Thinking. Equipes de design trabalham em conjunto com equipes de desenvolvimento na elaboração da documentação.	Materializados nos documentos gerados pelo Design Thinking. São complementados pelas equipes de desenvolvimento, no padrão da Engenharia de Requisitos, somente se existe exigência contratual do cliente e nestes casos as equipes de design auxiliam na complementação da documentação.	Logo após a conclusão das sessões de Design Thinking.
<b>B</b>	Materializados nos documentos gerados pelo Design Thinking. Somente as equipes de design trabalham na elaboração da documentação.	Materializados nos documentos gerados pelo Design Thinking e pelas histórias do usuário, elaborados pela própria equipe de design, durante as sessões do Design Thinking.	Logo após a conclusão das sessões de Design Thinking.
<b>C</b>	Materializados nos documentos gerados pelo Design Thinking. Somente as equipes de design trabalham na elaboração da documentação.	São gerados, após o término das sessões de Design Thinking, por uma equipe de analistas de requisitos nos padrões da Engenharia de Requisitos.	Logo após a conclusão das especificações de requisitos de sistema pelas equipes de analistas de requisitos.

Com base nas análises realizadas nos resultados do ponto de análise PA-05, pode-se afirmar que o Design Thinking está promovendo algum nível de documentação de requisitos de software, fato este observado nas três organizações. Um ponto importante é que na Organização A, a equipe de desenvolvimento participa das sessões de Design Thinking auxiliando a documentação dos artefatos, para evitar problemas futuros no desenvolvimento. Já na Organização B a documentação para o desenvolvimento de software é exclusivamente produzida pelo Design Thinking, o que demonstra uma substituição do processo de Engenharia de Requisitos (VERTELLI et al., 2013). Na organização C, como o foco do Design Thinking é o tratamento de requisitos de usuário, faz-se necessária a complementação dos requisitos de sistema por uma equipe de analistas de requisitos para o posterior início das fases de desenvolvimento de software.

### **PONTO DE ANÁLISE: PA-06 - Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos**

O conceito de apoio que estabelece que o Design Thinking vai além da elicitação de requisitos (CARROLL; RICHARDSON, 2016) permitiu fundamentar o

ponto de análise “PA-06 Compreender a aplicação do Design Thinking na etapa de Validação de Requisitos”, cujo objetivo é verificar se as atividades de validação de requisitos estão ocorrendo dentro do Design Thinking.

Foi possível verificar que em todas as organizações ocorre uma etapa formal de validação de requisitos com os usuários dentro do Design Thinking, com o objetivo de verificar consistência e completude, assim como ocorre na Engenharia de Requisitos. Nestas organizações, assim como também demonstrado pelos estudos de (HEHN; UEBERNICKEL, 2018; SOUZA; SILVA, 2015), a validação ocorre predominantemente por meio da análise dos protótipos desenvolvidos. Na Organização A, a validação dos requisitos identificados se dá principalmente pelos protótipos gerados, buscando verificar se os requisitos estão corretamente endereçados e atendem as necessidades do negócio. Na Organização B a validação ocorre a todo momento (INAYAT et al., 2014) pelo cliente e pelas próprias equipes de desenvolvimento, durante as sessões de Design Thinking. Nesta organização a validação iterativa tem por objetivo evitar eventuais divergências no produto que está sendo projetado uma vez que as estórias de usuário são produzidas pelas equipes de design. Já na Organização C, a equipe do design se reúne ao final do processo com todos os participantes das sessões de Design Thinking, para conduzir uma validação final dos documentos e protótipos gerados, com o intuito de validar se todos os requisitos do produto foram idealizados.

Com base nas análises realizadas nos resultados do ponto de análise PA-06, pode-se afirmar que o Design Thinking está promovendo algum nível de validação de requisitos de software, fato este observado nas três organizações.

### **PROPOSIÇÃO: P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos**

Após a análise dos resultados de cada um dos pontos de análise desta proposição, das respectivas citações e dos estudos de caso, pode-se concluir que a proposição “P1 - O Design Thinking é utilizado nas atividades da Engenharia de Requisitos” é verdadeira. Conforme descrito, foram encontradas evidências de que todas as etapas do processo de Engenharia de Requisitos estão sendo conduzidas de alguma forma pelas sessões de Design Thinking.

Percebe-se também a contribuição desta combinação de abordagens para a identificação da melhor solução para os problemas tratados, com um tratamento mais

efetivo de requisitos de usuário. As sessões de Design Thinking têm sido aplicadas para melhorar o entendimento dos problemas presentes no espaço do problema e a idealização da melhor solução de software para estes problemas, por meio de um processo participativo e criativo com os usuários da solução. Mesmo que o Design Thinking esteja auxiliando a Engenharia de Requisitos nestas organizações, percebe-se a necessidade de que as equipes de desenvolvimento de software estejam presentes em algum momento durante as sessões de Design Thinking. A presença destas equipes é necessária, seja para identificar os requisitos de sistema, analisar os requisitos identificados, auxiliar na documentação de requisitos e/ou para validar os requisitos sob a ótica de viabilidade tecnológica do desenvolvimento de software. Isso sugere, nestas organizações, que mesmo o Design Thinking trazendo uma abordagem nova para o tratamento de requisitos, ele em si não substituiu a Engenharia de Requisitos como um todo.

### 5.1.2 Proposição 2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis

O Quadro 20 descreve todos os conceitos de apoio utilizados para a definição da proposição P2 e a verificação ou não da identificação dos pontos de análise desta proposição no estudo de caso de cada uma das organizações investigadas:

**Quadro 20 – Síntese da proposição P2 nas organizações, fonte (O autor, 2020)**

P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis			
<p>Conceitos de apoio à proposição:</p> <ul style="list-style-type: none"> <li>• Enquanto o Design Thinking e as práticas ágeis foram concebidas para propósitos distintos, nós podemos identificar que ambos têm por objetivo entregar soluções rápidas com um atendimento parcial de uma necessidade particular (CORRAL; FRONZA, 2018).</li> <li>• Em comparação ao Design Thinking o ágil tem muitos paralelos: centrado no usuário, processo de desenvolvimento iterativo e de aprendizado e uma extensiva comunicação como time o que nos sugere que a metodologia do Design Thinking já foi introduzida para o desenvolvimento de TI (LINDBERG; MEINEL, 2011).</li> <li>• Há registros de que os métodos ágeis podem combinar com o Design Thinking para aprimorar o processo de desenvolvimento. Como ambos, desenvolvimento ágil e Design Thinking, dão ênfase nas pessoas em detrimento ao processo, ambos parecem se integrar naturalmente (PALACIN-SILVA et al., 2017).</li> </ul>			
Pontos de análise	Organização		
	A	B	C
PA-01 Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking	☺	☺	☹
<b>Legenda:</b>			
☺ Ponto de análise foi <b>amplamente</b> identificado na organização			
☺ Ponto de análise foi identificado, mas de forma <b>parcial</b> na organização			
☹ Ponto de análise <b>não foi identificado</b> na organização			

### **PONTO DE ANÁLISE: PA-01 - Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking**

Apesar dos estudos de (HEHN; UEBERNICKEL, 2018) e (VETTERLI et al. 2013) apresentarem a combinação da utilização do Design Thinking com o modelo de desenvolvimento de software baseado em métodos ágeis, ao analisar o ponto de análise “PA-01 Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking nas organizações deste estudo” foi identificado que nem sempre esta combinação está presente. A Organização A tem preferência em adotar o Design Thinking com métodos ágeis, entretanto a decisão de qual processo de desenvolvimento será utilizado é uma decisão exclusiva do cliente, e nesta organização já foi aplicado o Design Thinking com o modelo waterfall. A Organização B, por outro lado, só utiliza a combinação do Design Thinking com o método ágil e a Organização C utiliza exclusivamente o Design Thinking com o modelo de desenvolvimento de software *waterfall*, planejando uma transição para o método ágil nos próximos meses.

Para (CORRAL; FRONZA, 2018) a combinação do Design Thinking com os métodos ágeis propicia ondas de descobertas mais rápidas para o atendimento de soluções parciais de um determinado problema. Nesta pesquisa foi possível observar que na Organização A, a relação do Design Thinking com os métodos ágeis promove uma esteira cíclica e contínua de entregas para os usuários, tanto de análise dos problemas como também de entregas de desenvolvimento de software. Nesta organização as equipes de Design Thinking vão disponibilizando as entregas de entendimento do problema que está sendo tratado e a respectiva proposta de solução por meio de um protótipo, e as equipes ágeis já começam a criar a solução de software com base nestes produtos. Paralelamente, as equipes de design avançam para o próximo ciclo de aprofundamento e entendimento de outra parte do problema, com o intuito de gerar novas especificações, as quais servirão como base para o próximo ciclo de desenvolvimento de software das equipes ágeis. O objetivo é fazer com que o cliente final da solução experimente o quanto antes o produto para que possa repriorizar as frentes de Design Thinking.

Observa-se ainda que esta combinação tem grande efetividade, uma vez que as equipes de desenvolvimento de software, cujo foco é abordar a viabilidade tecnológica do problema, têm uma participação ativa com as equipes de design, cujo

foco é compreender as necessidades dos clientes. Esta sinergia entre as duas áreas corrobora com o estudo de (LINDBERG; MEINEL, 2011), no qual existem vários paralelos na combinação do Design Thinking com métodos ágeis, quais sejam: centrado no usuário, processo iterativo e de aprendizado e uma extensa comunicação entre as equipes das áreas de design e de desenvolvimento.

Para (PALACIN-SILVA et al., 2017) a combinação do Design Thinking com métodos ágeis pode propiciar um aprimoramento do processo de desenvolvimento. Na Organização B foi observada este aprimoramento por meio do tratamento de problemas complexos com inovação já no início de concepção dos projetos de software. O cliente desta organização relatava muitas dificuldades em se criar novos produtos, baseados em software, para as áreas de negócio. A inserção do Design Thinking antes do início dos ciclos de desenvolvimento ágil, permitiu que o cliente pudesse realizar um entendimento dos seus problemas, na grande maioria complexos, e propusesse soluções inovadoras para estes problemas as quais eram desenvolvidas pelos ciclos de entregas do método ágil.

### **PROPOSIÇÃO: P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis**

Uma conclusão que pode-se chegar sobre as observações dos estudos de caso nestas organizações, é de que o método de desenvolvimento de software utilizado não tem influência na efetividade e/ou viabilidade da aplicação do Design Thinking com a Engenharia de Requisitos no contexto destas empresas, pois como observado a Organização A utiliza métodos ágeis ou o modelo waterfall, a Organização B utiliza exclusivamente métodos ágeis e a Organização C utiliza exclusivamente o modelo waterfall.

Mesmo assim, percebe-se que diante dos resultados do ponto de análise, dos estudos de caso e citações é possível afirmar que a proposição “P2 - O Design Thinking é utilizado com projetos que adotam os métodos ágeis” é verdadeira, pois foram encontradas evidências desta utilização nas Organizações A e B. Por outro lado, mesmo a Organização C ainda não estar utilizando esta combinação, enfatizou que nos próximos meses os processos da organização estarão migrando para o método ágil, o que fará com que esta combinação passe a ser identificada.



### 5.1.3 Proposição 3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis

O Quadro 21 descreve todos os conceitos de apoio utilizados para a definição da proposição P3 e a verificação ou não da identificação dos pontos de análise desta proposição no estudo de caso de cada uma das organizações investigadas:

**Quadro 21 – Síntese da proposição P3 nas organizações, fonte (O autor, 2020)**

P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis			
Conceitos de apoio à proposição: <ul style="list-style-type: none"> <li>• Como aprimorar a engenharia de requisitos para incorporar os recursos úteis das metodologias ágeis de forma a suportar o desenvolvimento de <i>apps</i>, em consonância com o desenvolvimento de grandes sistemas. Nós precisamos de um método que combina o melhor das duas abordagens: Design Thinking e Engenharia de Requisitos. (VETTERLI et al., 2013).</li> </ul>			
Pontos de análise	Organização		
	A	B	C
PA-01 Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking	☹️	😊	☹️
<b>Legenda:</b> <ul style="list-style-type: none"> <li>😊 Ponto de análise foi <b>amplamente</b> identificado na organização</li> <li>☹️ Ponto de análise foi identificado, mas de forma <b>parcial</b> na organização</li> <li>⊗ Ponto de análise <b>não foi identificado</b> na organização</li> </ul>			

#### **PONTO DE ANÁLISE: PA-01 - Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking**

Alguns estudos sugeriram a aplicação do Design Thinking para o desenvolvimento de aplicativos móveis (VETTERLI et al. 2013; SOUZA; SILVA, 2015). No estudo de (SOUZA; SILVA, 2015) o Design Thinking foi aplicado para auxiliar a elicitación de requisitos de uma aplicação de mobile-learning, entretanto não fica claro se esta escolha foi uma decisão baseada na família de software. Já (VETTERLI et al. 2013) descrevem que a utilização do Design Thinking é chave de sucesso para o desenvolvimento de aplicativos móveis, pois aprimora o processo tradicional de Engenharia de Requisitos. Contudo, ao analisar o ponto de análise “PA-01 Identificar as motivações e o ambiente das iniciativas que adotaram o Design Thinking nos estudos de caso desta pesquisa”, identificou-se que nem sempre o Design Thinking está sendo utilizado em aplicações móveis. Os resultados demonstram que o Design Thinking está aprimorando e até algumas vezes substituindo a Engenharia de Requisitos, mas esta situação ocorre não necessariamente quando aplicado no desenvolvimento de uma família específica de software.

Para (VETTERLI et al., 2013) a aplicação do Design Thinking com a Engenharia de Requisitos seria a maneira mais efetiva de tratar requisitos em sistemas móveis,

porém conforme observado nesta pesquisa nem sempre a família de software tem influência na decisão desta aplicação. Entretanto, um ponto em comum com o estudo de (VETTERLI et al., 2013) é que em todas as organizações foi possível observar que a aplicação do Design Thinking visava atender à necessidade de se ajustar fortemente às expectativas dos usuários. Conforme observado nos estudos de caso, a Organização A aplica o Design Thinking, em sua grande maioria em softwares desktop, mas observa um crescimento de demanda para a utilização em aplicações móveis. A Organização B aplica o Design Thinking tanto em softwares desktop quanto em aplicações móveis, uma vez que o produto de software desenvolvido por esta organização é composto pelas duas famílias de software. Já a Organização C aplica o Design Thinking na grande maioria das vezes em softwares desktop, com poucas experimentações em aplicações móveis, uma vez que nessa organização a utilização de aplicações móveis é muito limitada devido à natureza do negócio.

### **PROPOSIÇÃO: P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis**

No estudo descrito por (VETTERLI et al. 2013) a natureza de aplicações móveis exige uma abordagem mais rápida e efetiva para o tratamento dos requisitos de software, mas com base nos estudos de caso conduzidos por esta pesquisa e no contexto das organizações investigadas, constatou-se que aplicações de grande porte, muitas vezes baseadas exclusivamente em softwares desktop, estão utilizando o Design Thinking. Esta utilização ocorre tanto para auxiliar o desenho da solução como também para auxiliar o levantamento de requisitos de software. Isso leva à conclusão de que nestas organizações, a família de software que está sendo desenvolvida não tem uma influência na escolha da utilização do Design Thinking. Percebe-se que esta escolha está muito mais ligada às características dos projetos como: natureza vaga do problema, problemas complexos, necessidade de inovação dentre outras, do que por uma família de software em específico.

Após os resultados do ponto de análise, dos estudos de caso e citações é possível afirmar que a proposição “P3 - O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis” é verdadeira, pois foram encontradas evidências desta utilização nas Organizações A, B e C.

## **5.2 Generalização dos Estudos de Caso**

Como esta pesquisa analisou estudos de caso em empresas brasileiras com características diversas como o porte e segmento de atuação, e mesmo com objetivos distintos na aplicação do Design Thinking com o desenvolvimento de software, ressalta-se que não é possível assegurar a generalização das constatações deste trabalho para outras organizações.

## **5.3 Considerações sobre o capítulo**

Este capítulo apresentou a análise consolidada de cada uma das proposições com base nos resultados dos estudos de caso. Para dar luz à questão primordial desta pesquisa, foram analisados cada um dos pontos de análise e o resultado destas análises em cada uma das proposições, correlacionando estas análises com estudos anteriores.

## CAPÍTULO 6 - CONSIDERAÇÕES FINAIS

O objetivo deste capítulo é apresentar a conclusão desta pesquisa, a relevância deste estudo, sua contribuição e limitações e trabalhos futuros.

### 6.1 Análise da Questão de Pesquisa Principal

O tema deste trabalho de pesquisa é tentar responder à questão de pesquisa principal: ***Como o Design Thinking auxilia a Engenharia de Requisitos em projetos de software?***

A resposta à questão de pesquisa principal, com o conjunto de proposições associadas, tem por objetivo principal **compreender como as práticas do Design Thinking são aplicadas na Engenharia de Requisitos em projetos de software**, através dos objetivos específicos:

- I - Identificar as práticas do Design Thinking que estão sendo utilizadas na Engenharia de Requisitos; e
- II - Analisar as práticas identificadas sob a ótica da Engenharia de Requisitos.

É possível concluir que objetivos foram alcançados, pois com a construção do referencial teórico, a análise e discussão de cada um dos pontos de análise nas organizações foi possível estabelecer um panorama, mesmo que restrito pela quantidade de organizações avaliadas, de como tem ocorrido a interação do Design Thinking com a Engenharia de Requisitos. Esta combinação demonstra que ocorre uma extensão da área de engenharia de software com a adoção dos princípios de cocriação oriundos de uma área de design.

Para responder à questão primordial de pesquisa verificou-se que o Design Thinking, nas organizações analisadas, tem se mostrado como um método eficiente para o tratamento de problemas complexos, para geração de soluções de software inovadoras e para identificação de requisitos de software, dentro dos domínios do problema e da solução presentes na Engenharia de Requisitos. Mesmo tendo um foco mais voltado para requisitos de usuário, apesar de algumas organizações já estarem especificando requisitos de sistema funcionais e/ou não-funcionais, observa-se uma

contribuição para as equipes de desenvolvimento. Estas equipes ao receberem um protótipo de software e documentos complementares, que representam um problema amplamente discutido, ampliam a efetividade do desenvolvimento de software. Por último, verificou-se que a contribuição do Design Thinking não ocorre somente na etapa de elicitação de requisitos, mas também nas etapas de análise e negociação, documentação e validação de requisitos as quais ocorrem dentro do processo de Engenharia de Requisitos.

## **6.2 Relevância do estudo**

Conforme demonstrado no Capítulo 1 desta dissertação, pouco se conhece sobre os efeitos da combinação do Design Thinking com a Engenharia de Requisitos. Os estudos encontrados nesta área têm um caráter experimental da aplicação do Design Thinking em projetos de software, sem analisar o impacto desta aplicação sob o ponto de vista da Engenharia de Requisitos. Dessa forma esta pesquisa visa trazer um aprofundamento, por meio da análise de estudos de caso, de como cada uma das etapas da Engenharia de Requisitos estão sendo auxiliadas pela utilização do Design Thinking em projetos de software. Os resultados encontrados nos pontos de análise, os quais foram integralmente avaliados em cada um dos estudos de caso, foram analisados e discutidos individualmente para cada uma das etapas do processo de Engenharia de Requisitos.

## **6.3 Contribuições da pesquisa**

Esta pesquisa apresenta como contribuição da pesquisa uma compreensão de como as técnicas do Design Thinking auxiliam a Engenharia de Requisitos, apoiando a utilização dessas técnicas e práticas em outras organizações que desejem combinar o uso do Design Thinking no desenvolvimento de software. Além disso, este trabalho poderá auxiliar outras pesquisas que possam surgir a partir deste tema.

Os resultados encontrados nesta pesquisa demonstram a inter-relação entre os métodos utilizados pelo Design e que estão sendo utilizados pela Engenharia de Software, trazendo uma melhor compreensão desta relação sob a ótica do processo de Engenharia de Requisitos. Tais resultados poderão auxiliar outras organizações na compreensão, implementação e limitação das práticas do Design Thinking combinadas com a Engenharia de Requisitos em projetos de software.

## **6.4 Limitações**

Como esta pesquisa realizou estudos de caso em empresas brasileiras que desenvolvem software, com características de atuação distintas, não foi possível a generalização deste estudo para todas as organizações que se utilizam da combinação do Design Thinking em projetos de software, uma vez que as características peculiares de outras organizações e de outros países não foram observadas e conseqüentemente analisadas.

Também não foi possível validar a efetividade do protocolo estabelecido por esta pesquisa, devido ao conjunto limitado de estudos de caso realizados.

## **6.5 Trabalhos futuros**

Como trabalhos futuros pode-se sugerir uma ampliação de estudos de caso, considerando empresas de desenvolvimento de software em um mesmo segmento de atuação. Isso permitirá uma melhor compreensão e aprofundamento da combinação do Design Thinking com a Engenharia de Requisitos.

Também pode-se considerar como um trabalho futuro a compreensão dos motivos que levam as organizações a utilizar o Design Thinking (ou qualquer outra metodologia de tratamento de problemas para geração de soluções inovadoras) para realizar o tratamento de requisitos de negócio e/ou requisitos de usuário. Da mesma forma seria possível avaliar se esta escolha está ligada à necessidade de inovação presente nas soluções de software atuais, e que eventualmente não podem ser atendidas pelo processo tradicional de Engenharia de Requisitos.

Um outro estudo poderia aprofundar no entendimento de quais técnicas do Design Thinking estão sendo mais utilizadas pelas organizações, e se a escolha destas técnicas tem alguma relação com o processo de Engenharia de Requisitos de software.

Por fim, um outro trabalho futuro poderia endereçar as motivações da utilização do Design Thinking em projetos de software. Algumas observações coletadas nesta pesquisa sugerem uma maior assertividade no negócio e inovação para problemas complexos, entretanto não foi realizado nenhum aprofundamento quanto aos critérios e motivadores para esta adoção.

## REFERÊNCIAS

- (ADIKARI et al., 2013) ADIKARI, S.; MCDONALD, C.; CAMPBELL, J. **Reframed contexts: Design Thinking for Agile User Experience Design**. In: International Conference of Design, User Experience and Usability, p. 3-12, 2013.
- (ARAÚJO et al., 2015) ARAÚJO, R.; ANJOS, E.; SILVA, D.R. **Trends in the use of Design Thinking for Embedded Systems**. In: 15th International Conference on Computational Science and Its Applications, 2015.
- (BABOK, 2011) BABOK. **O Guia para o Corpo de Conhecimento de Análise de Negócios (Guia BANOK)**. IIBA-International Institute of Business Analysis, 2011.
- (BAKER et al., 2019) BAKER, C.; DENG, L.; CHAKRABORTY, S.; DEHLINGER, J. **Automatic Multi-class Non-Functional Software Requirements Classification Using Neural Networks**. In: IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019.
- (BATOOL et al., 2013) BATOOL, A.; MOTLA, Y.H.; HAMID, B.; ASGHAR, S.; RIAZ, M.; MUKHTAR, M.; AHMED, M. **Comparative Study of Traditional Requirement Engineering and Agile Requirement Engineering**. In: 15th International Conference on Advanced Communications Technology (ICACT), 2013.
- (BROWN, 2008) BROWN, T. **Design Thinking**. Harvard Business Review, v. 86(5), p. 84-92, 2008.
- (BUDD, 2004) BUDD, J. W. **Mind Maps as Classroom Exercises**. The Journal of Economic Education, v. 35, p. 35-46, 2004.
- (CARROLL; RICHARDSON, 2016) CARROLL, N.; RICHARDSON, I. **Aligning Healthcare Innovation and Software Requirements through Design Thinking**. In: International Workshop on Software Engineering in Healthcare Systems, 2016.
- (CHUNG et al., 2000) CHUNG, L.; NIXON, B.A.; Yu, E.; MYLOPOULOS, J. **Non-Functional Requirements in Software Engineering**. Springer, 2000.
- (CORRAL et al., 2013) CORRAL, L.; SILLITTI, A.; SUCCI, G. **Software Development Process for Mobile Systems – Is Agile Really Taking Over the Business?** In: 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS), 2013.
- (CORRAL; FRONZA, 2018) CORRAL, L.; FRONZA, I. **Design Thinking and Agile Practices for Software Engineering: An Opportunity for Innovation**. In: 19<sup>th</sup> Annual Conference on Information Technology Education (SIGITE), 2018.
- (CURCIO et al., 2018) CURCIO, K.; NAVARRO, T.; MALUCELLI, A.; REINEHR, S. **Requirements engineering: A systematic mapping study in agile software development**. The Journal of System and Software, v. 139 p. 32-50, 2018.

(DEHLINGER; DIXON, 2011) DEHLINGER, J.; DIXON, J. **Mobile application software engineering: Challenges and research directions**. In: Workshop on mobile software engineering, v. 2, p. 29-32, 2011.

(EISENHARDT, 1989) EISENHARDT, K. M. **Building Theories from Case Study Research**. The Academy of Management Review, v. 14, n. 4, p. 532-550, 1989.

(HEHN; UEERNICKEL, 2018) HEHN, J.; UEERNICKEL, F. **The Use of Design Thinking for Requirements Engineering An Ongoing Case Study in the Field of Innovative Software-Intensive Systems**. In: 26th IEEE International Requirements Engineering Conference, 2018.

(HEHN et al., 2020) HEHN, J.; MENDEZ, D.; UEERNICKEL, F.; BRENNER, W.; BROY, M. **On Integrating Design Thinking for a Human-centered Requirements Engineering**. IEEE Software, v. 37, Issue 2, 2020.

(IEEE, 1993); **IEEE Std 830-1993 - IEEE Guide to Software Requirements Specifications**. Software Engineering Standards, IEEE Computer Science Press, 1993.

(INAYAT et al., 2014) INAYAT, I.; SALIM, S.S.; MARCZAK, S.; DANEVA, M.; SHAMSHIRBAND, S. **A systematic literature review on agile requirements engineering practices and challenges**. Computer in Human Behavior, v. 51, p. 915-929, 2014.

(JALIL et al., 2019) JALIL, R.; KHALID, J.; MARYAM, M.; KHALID, M.; CHEEMA, S.N.; IQBAL, I. **Requirements Elicitation for Bespoke Software Development: A Review Paper**. In: The International Conference on Intelligent Technologies and Applications, Springer CCIS 932, p. 805-821, 2019.

(JARZEBOWICZ; POLOCKA, 2017) JARZEBOWICZ, A.; POLOCKA, K. **Selecting Requirements Documentation Techniques for Software Projects: a Survey Study**. In: 1<sup>st</sup> International Conference on Lean and Agile Software Development FedCSIS, 2017.

(KARLSON; RYAN, 1997) KARLSON, J.; RYAN, K. **A cost-value approach for prioritizing requirements**. IEEE Software, v. 14, Issue: 5, 1997.

(KLINKMKÜLLER et al., 2010) KLINKMKÜLLER, C.; LUDWIG, A.; FRANCYK, B.; KLUGE, R. **Visualizing Business Capabilities in the Context of Business Analysis**. In: Abramowicz W., Tolksdorf R, Business Information Systems, Lecture Notes in Business Information Processing, v 47, Springer, 2010.

(KOTONYA; SOMMERVILLE, 1998) KOTONYA, G.; SOMMERVILLE, I. **Requirements engineering: processes and techniques**. J. Wiley & Sons, 1998.

(KUMAR et al., 2019) KUMAR, S.V.; KASIVISWANATH, N.; HARITHA, M. **Software Engineering: Requirements Elicitation Techniques, Issues and Challenges in Industry**. S. Vinay Kumar Journal of Engineering Research and Application, v. 9, Issue 2 (Series -I), p 10-15, 2019.



(LABONTE et al., 1999) LABONTE, R.; FEATHER, J.; HILLS, M. **A story/dialogue method for health promotion knowledge development and evaluation**. Health Education Research, v.14, p. 39-50, 1999.

(LINDBERG et al., 2011) LINDBERG, T.; MEINEL, C.; WAGNER, R. **Design Thinking: A Fruitful Concept for IT Development?** Design Thinking: Understand – Improve – Apply, p. 3-18, Springer, 2011.

(MEDEIROS et al., 2015) MEDEIROS, J.D.R.V.; ALVES, D.C.P.; VASCONCELOS, A.; SILVA, C.; WANDERLEY, E. **Requirements engineering in agile projects: A systematic mapping based in evidences of industry**. In: XVIII Ibero-American Conference on Software Engineering (CIBSE)/ESELAW, 2015.

(NUSEIBEH; EASTERBROOK, 2000) NUSEIBEH, B.; EASTERBROOK, B. **Requirements Engineering: a roadmap**. In: International Conference on Software Engineering '00, p 35-46, ACM, 2000.

(PAETSCH et al., 2003) PAETSCH, F.; EBERLEIN, A.; MAURER, F. **Requirements Engineering and Agile Software Development**. In: Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.

(PALACIN-SILVA et al., 2017) PALACIN-SILVA, M.; KHAKUREL, J.; HAPPONEN, A.; HYNINEN, T.; PORRAS, J. **Infusing Design Thinking Into a Software Engineering Capstone Course**. In: The 30<sup>th</sup> IEEE Conference on Software Engineering Education and Training, 2017.

(PAULA; CORMICAN, 2016) PAULA, D.; CORMICAN, K.; **Understanding Design Thinking in Design Studies (2006-2015): A Systematic Mapping Study**. In: 14<sup>th</sup> International Design Conferences, p. 1-10, 2016.

(PMI GUIDE, 2015) **Business analysis for practitioners. A practice guide**. Project Management Institute, 2015.

(RAMINGWONG, 2012) RAMINGWONG, L. **A Review of Requirements Engineering Process, Problems and Models**. In: International Journal of Engineering Science and Technology, v. 4, n. 6, 2012.

(REINEHR, 2008) REINEHR, S.S. **Reuso Sistematizado de Software e Linhas de Produto de Software no Setor Financeiro: Estudos de Caso no Brasil**. Tese (Doutorado), Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil, 2008.

(ROMAN, 1985) ROMAN, G.C. **A Taxonomy of Current Issues in Requirements Engineering**. Computer, v. 18, n. 4, p. 14-23, 1985.

(SAMETZ; MAYDONEY, 2003) SAMETZ, R.; MAYDONEY, A. **Storytelling through design**. Design Management Journal (Former Series), v. 14(4), p.18-34, 2003.

(SHEN; LO, 2018) SHEN, X.; LO, CH. **Modern Service Design Thinking on Traditional Culture-Based Services: A Case Study of the Service Businesses in Suzhou Old Town Areas**. Cross-Cultural Design. Applications in Cultural Heritage,

Creativity and Social Development, CCD 2018. Lecture Notes in Computer Science, v. 10912, Springer, 2018.

(SIMON, 1996) SIMON, H.A. **The Sciences of the Artificial**. MIT Press, 1996.

(SOLEDADE et al., 2013) SOLEDAD, M.; FREITAS, S.P.; FANTINATO, R.S.; ARAUJO, U. **Experimenting with Design Thinking in Requirements Refinement for a Learning Management System**. In: Anais do Simpósio Brasileiro de Sistemas de Informação, 2013

(SOMMERVILLE, 2019) SOMMERVILLE, I. **Software engineering**. Pearson Education do Brasil, São Paulo , 2019.

(SOUZA et al., 2017) SOUZA, A.F.B.; FERREIRA, B.M.; CONTE, T. **Aplicando Design Thinking em Engenharia de Software: um Mapeamento Sistemático**. In: XX Ibero-American Conference on Software Engineering, 2017.

(SOUZA et al., 2018) SOUZA, A.F.B.; FERREIRA, B.; VALENTIM, N.; CONTE, T. **An Experience Report on Teaching Multiple Design Thinking Techniques to Software Engineering Students**. In: SBES, 2018.

(SOUZA; SILVA, 2015) SOUZA, C.L.C; SILVA, C. **An Experimental Study of the Use of Design Thinking as a Requirements Elicitation Approach for Mobile Learning Environments**. CLEI Eletronic Journal, v. 18 n. 1, 2015.

(VALASKI, 2017) VALASKI, J. **Derivação de Requisitos Funcionais a Partir de Descrições de Domínio**. Tese (Doutorado), Pontifícia Universidade Católica do Paraná, Curitiba, Brasil, 2017.

(VALENTIM et al., 2018) VALENTIM, J.M.C.; SILVA, W.; CONTE, T. **The Student's Perspective on Applying Design Thinking for the Design of Mobile Applications**. In: IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track, 2018.

(VETTERLI et al., 2013) VETTERLI, C.; BRENNER, W.; UEBERNICKEL, F., PETRIE, C. **From Palaces to Yurts – Why Requirements Engineering Needs Design Thinking**. IEEE Computer Society, 2013.

(WASSERMAN, 2010) WASSERMAN, A.I. **Software Engineering Issues for Mobile Application Development**. In: 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2010.

(WHITTLE et al., 2010) WHITTLE, J.; SAWYER, P.; BENCOMO, N.; CHENG, B. H. C.; BRUEL, J.M. **RELAX: a language to address uncertainty in self-adaptive systems requirement**. Requirements Engineering, v. 15, n. 2, p. 177-196, 2010.

(YIN, 2015) YIN, R. K. **Estudo de caso - planejamento e métodos**. 5a. ed. Porto Alegre: Bookman, 2015.

(ZANLORENCI; BURNETT, 1998) ZANLORENCI, E.; BURNETT, R.C. **Modelo para Qualificação da Fonte de Informação Cliente e de Requisito Funcional**. In: SBES98, XII Simpósio Brasileiro de Engenharia de Software WER'98, I Workshop de Engenharia de Requisitos, v. 1, n. 1, p. 39-48, 1998.

## APÊNDICE A – PROTOCOLO DE PESQUISA – TERMO DE CONFIDENCIALIDADE

Curitiba, [dia] de [mês] de 2019.

**[Empresa]**

**[Responsável]**

Prezado Senhor,

Este Termo de Confidencialidade visa estabelecer um acordo entre os pesquisadores VINICIUS PRODUCIMO, ANDREIA MALUCELLI E SHEILA REINEHR, doravante denominados Pesquisadores, e a Organização [EMPRESA], doravante denominado Organização Participante, a respeito da confidencialidade das informações coletadas durante o processo de pesquisa da tese de doutorado do primeiro, sob orientação do segundo.

Por meio deste Termo de Confidencialidade, os Pesquisadores se comprometem a:

- Portar-se com discrição em todos os momentos da pesquisa acadêmica, não comentando ou divulgando qualquer tipo de informação que tenha sido repassada de forma oral ou escrita.
- Não divulgar o nome da Organização Participante, em qualquer meio, a menos que expressamente autorizado por esta.
- Não divulgar, em qualquer meio, os dados e informações individualizados coletados durante o processo de pesquisa na Organização Participante.
- Divulgar, em formato de tese, artigos e apresentações, apenas os dados agregados, dos quais não se possa retirar ou inferir a identificação da Organização Participante.
- Retornar para a Organização Participante, em formato agregado, todos os dados de todos os estudos de caso conduzidos.

As assinaturas abaixo expressam a concordância quanto ao cumprimento deste Termo de Confidencialidade, por prazo indeterminado.

Andreia Malucelli, Dra.  
Programa de Pós-Graduação em Informática  
Pontifícia Universidade Católica do Paraná

Sheila Reinehr, Dra.  
Programa de Pós-Graduação em Informática  
Pontifícia Universidade Católica do Paraná

Vinicius Produçimo  
Programa de Pós-Graduação em Informática  
Pontifícia Universidade Católica do Paraná

## **APÊNDICE B – PROTOCOLO DE PESQUISA – APRESENTAÇÃO DA PESQUISA**

Curitiba, [dia] de [mês] de 2019.

[Nome da Empresa]

[Nome do Responsável]

[Prezado],

Venho, por meio desta, solicitar a sua autorização para a condução de um estudo de campo da dissertação de mestrado do aluno Vinicius Prodocimo que está sendo desenvolvida sob nossa orientação e co-orientação no Programa de Pós-Graduação em Informática da PUC-PR.

O objetivo principal da pesquisa é entender como as práticas do Design Thinking são aplicados nos processos de Engenharia de Requisitos pelos departamentos de Tecnologia de Informação de empresas brasileiras.

A pesquisa será realizada por meio de entrevista presencial, que visa coletar as informações necessárias para extrair resultados claros e concisos sobre como as práticas do Design Thinking são aplicadas nos processos de Engenharia de Requisitos em projetos de software.

Gostaria, ainda, de afirmar o nosso compromisso em relação à confidencialidade das informações prestadas. Todos os dados serão tratados de forma a preservar a privacidade, tanto dos entrevistados, quanto da organização. Nenhuma informação personalizada será publicada, a menos que autorizado formalmente pela organização. Um Termo de Confidencialidade será assinado pelos pesquisadores, com termos a critério da organização.

Agradecemos a colaboração e permanecemos integralmente à disposição.

Atenciosamente,

Orientadora: Prof. Dra. Andreia Malucelli

Co-orientadora: Prof. Dra. Sheila Reinehr,

Programa de Pós-Graduação em Informática

Pontifícia Universidade Católica do Paraná - PUCPR

## **APÊNDICE C – PROTOCOLO DE PESQUISA – VISÃO GERAL DA PESQUISA**

### **IDENTIFICAÇÃO:**

Projeto de pesquisa da dissertação de mestrado do estudante Vinicius Prodocimo, que está sendo desenvolvida sob orientação da Prof. Dra. Andreia Malucelli no Programa de Pós-Graduação em Informática da PUC-PR.

### **OBJETIVO DA PESQUISA:**

Compreender a aplicação do Design Thinking na Engenharia de Requisitos em projetos de software.

### **QUESTÃO QUE A PESQUISA VISA RESPONDER:**

Como o Design Thinking auxilia a Engenharia de Requisitos em projetos de software?

### **PÚBLICO ALVO:**

Organizações de TI do Brasil.

### **PROCEDIMENTOS OPERACIONAIS:**

Envio de convite via e-mail e contatos telefônicos para agendamento de entrevista presencial com o corpo técnico e gerencial das unidades de desenvolvimento de software.

### **CONFIDENCIALIDADE DAS INFORMAÇÕES:**

Nenhuma informação disponibilizada pelas organizações será divulgada de forma individualizada ou de maneira que seja possível caracterizar a organização. Todo o tratamento das informações irá ocorrer de forma agregada, mantendo o anonimato das organizações participantes.

### **PÚBLICO ALVO (PAPÉIS)**

Analistas de sistemas, profissionais de Design Thinking e gerentes de projetos que atuem diretamente no desenvolvimento de projetos de software.

**QUESTÕES E PROPOSIÇÕES**

Questão Primordial

*Como o Design Thinking auxilia a Engenharia de Requisitos em projetos de software?*

**P1**

O Design Thinking é utilizado nas atividades da Engenharia de Requisitos

**P2**

O Design Thinking é utilizado com projetos que adotam os métodos ágeis

**P3**

O Design Thinking é utilizado para apoiar o desenvolvimento de aplicativos móveis